




Phase transition in the bipartite z -matching

Till Kahlke^{1,a}, Martin Fränzle^{2,b}, and Alexander K. Hartmann^{1,c} 

¹ Institut für Physik, Carl von Ossietzky Universität Oldenburg, 26111 Oldenburg, Germany

² Department für Informatik, Carl von Ossietzky Universität Oldenburg, 26111 Oldenburg, Germany

Received 7 October 2021 / Accepted 16 November 2021 / Published online 17 December 2021
© The Author(s) 2021

Abstract. We study numerically the maximum z -matching problems on ensembles of bipartite random graphs. The z -matching problem describes the matching between two types of nodes, *users* and *servers*, where each server may serve up to z users at the same time. Using a mapping to standard maximum-cardinality matching, and because for the latter there exists a polynomial-time exact algorithm, we can study large system sizes of up to 10^6 nodes. We measure the capacity and the energy of the resulting optimum matchings. First, we confirm previous analytical results for bipartite regular graphs. Next, we study the finite-size behaviour of the matching capacity and find the same scaling behaviour as before for standard matching, which indicates the universality of the problem. Finally, we investigate for bipartite Erdős–Rényi random graphs the saturability as a function of the average degree, i.e. whether the network allows as many customers as possible to be served, i.e. exploiting the servers in an optimal way. We find phase transitions between unsaturable and saturable phases. These coincide with a strong change of the running time of the exact matching algorithm, as well with the point where a minimum-degree heuristic algorithm starts to fail.

1 Introduction

Phase transitions in combinatorial optimisation or in constraint-satisfaction problems [1–4] have been an active area of research at the interface of statistical mechanics and computer science since more than 2 decades. Usually, non-deterministically polynomial (NP) complete [5] or NP-hard problems are studied, i.e. problems for which so far no algorithm is known which runs in the worst case in polynomial time as a function of the system size. Thus, so far only worst-case exponential-time algorithms are available, i.e. the problems are *hard*. On the other hand, problems running in polynomial (P) time are often termed *easy*. But since it is not known, i.e. proven, whether P forms a proper subclass of NP, or whether maybe $P=NP$, one has been interested since almost the beginning of computer science in finding out what makes a problem hard. Among many other approaches, also numerical experiments and statistical-mechanics calculations have been performed. For this purpose, ensembles of random problems have been considered, which are hard in the worst case but for some regions in parameter space typically require only a polynomial running time, meaning they are typically easy there. Here in particular phase transitions [1, 6], e.g. with respect to the solvability have been

observed when varying suitable ensemble parameters. These phase transitions often coincide with changes of the typical complexity from easy to hard. Thus, the structure of problems from such ensembles, in particular near phase transitions, may teach us about the source of computational hardness. Such phase transitions have been studied, e.g. for constraint-satisfaction problems such as satisfiability (SAT) [7, 8] or colouring [9]. In addition, optimisation problems such as the travelling salesperson [10–13], vertex cover [14–17] or number partitioning [18, 19] have been investigated. Beyond delivering insight into the structure of problems, this research performed at the interface of physics and computer science has also led to algorithmic advances like the development of efficient message-passing algorithms as Belief Propagation or Survey Propagation [20].

Nevertheless, not only hard optimisation or constraint-satisfaction problems may exhibit changes of problem space structure and corresponding changes of the computational complexity. In addition, ensembles of polynomially solvable problems such as shortest paths, maximum flows or graph matching may be of interest and show corresponding phase transitions. Usually, in physics, such algorithms are used to investigate models like random magnets [21, 22]. Here, we want to perform a fundamental study of such a phase transition for a generalisation of the graph matching problem. As we will see here, we observe changes of the algorithmic behaviour related to this phase transition. This shows that such a coincidence of the change of a suitably defined solvability and of algorithmic complexity

^a e-mail: till.kahlke@uol.de

^b e-mail: martin.fraenzle@informatik.uni-oldenburg.de

^c e-mail: a.hartmann@uni-oldenburg.de

(corresponding author)

exists also for a polynomially solvable problem and can, therefore, easier be studied numerically.

For a given graph, the maximum-cardinality matching problem, also just called matching, considers subsets of edges, such that each node is incident to at most one edge in the subset and such that the cardinality of matching is maximum. This problem is widely studied in computer science usually from the algorithmic point of view [23] with the aim to find efficient algorithms. But also in the field of statistical mechanics it has already played its role, as it was among the first optimisation problems studied, and therefore, it has inspired the field a lot. First, the model was solved analytically using a replica-symmetric approach [24, 25] on bipartite random graphs with random edge weights, i.e. for the maximum-weight matching instead of the maximum-cardinality matching. This suitability of a replica-symmetric calculation means that the thermodynamic behaviour of matching for this ensemble is not very complex, similar to a ferromagnet. Later, the solution was extended to arbitrary graphs and the finite-size behaviour of the matching capacity, i.e. the sum of the weights of the edges in the matching, was obtained [26]. In addition, studied were Euclidean variants [27–29]. The case where more than two nodes are connected per matching element was also considered with a statistical mechanics approach [30]. Furthermore, so called *dimer coverings*, i.e. perfect matchings involving all nodes, on d -dimensional lattices without edge weights were studied [31]. Since no energy is involved, the number of matchings, i.e. the entropy, was mainly studied [32–34]. Such studies of entropies of dimer coverings were extended also to include energy for the edges [35], or to mixtures of dimers and single atoms [36]. In addition, entropies of the matchings [37] or dimer coverings [38] were considered for various more general random graph structures.

In this work, we study a phase transition of the satisfiable–unsatisfiable type for the z -matching problem, which is a generalisation of the standard matching. The model describes a set of N users and a set of S servers, possible user-server connections are described by a bipartite graph. Each user shall be served by one server, while each server may serve up to z users at the same time. The system is characterised by its capacity, i.e. the number of users which can be served simultaneously. An example for the application of z -matching are wireless communication networks [39, 40]. Our study is motivated by a previous work of Kreačić and Bianconi [41], who have studied, to our knowledge for the first time in statistical mechanics, the z -matching problem analytically with the approximate cavity approach and numerically with a message-passing algorithm. They have obtained the capacity of the system for two ensembles, namely for fixed degree and Poissonian bipartite graphs. They showed that for both cases, parameter combinations exists, where the capacity converges to its maximum possible value, i.e. a *saturable* phase, when increasing the average node degree.

Here, we expand on this work using an exact numerical matching algorithm. Since this algorithm allows

for calculating exact optimum matchings in polynomial time, we are able to solve exactly very large graphs of up to $N = 500000$ users. To start, we confirm with our exact numerical approach the previously obtained analytical results from Ref. [41]. In addition, we find the same finite-size scaling behaviour of the capacity as for standard matching [26]. In the main part of our work, for the case of the Poissonian random graphs with average user degree k , we investigate the model with respect to the phase transition between saturability and unsaturability for some typical parameter combinations of z and the ratio N/S . We determine with high precision the phase-transition point k_c using finite-size scaling techniques. In addition, we obtain the critical exponent ν characterising the phase transition. By analysing the run time of the exact algorithm and furthermore studying an approximation algorithm, we are able to show that the phase transition coincides with remarkable changes of the algorithmic performance.

The remainder of the paper is organised as follows: next, we define the z -matching problem and the measurable quantities we have evaluated, together with the ensemble of random graphs we have studied. In the third section, we present the methods we have applied. In the main section, we present our results, and finally, we summarise our work and outline further research directions.

2 Model

2.1 z -matching

We consider bipartite graphs $G = (V, E)$ with vertices $i \in V = A \cup B$ consisting of $N = |A|$ users and $S = |B|$ servers. We denote the ratio of number of users and servers by

$$\eta = N/S. \quad (1)$$

Since the graph is bipartite, each edge connects a user and a server. This means that the average degree k of the users and the average degree q of the servers are related by

$$Nk = Sq. \quad (2)$$

When inserting Equation 1, it follows that $q = \eta k$.

A z -matching M_z is a subset of E in which each user is adjacent to at most one server and each server is adjacent to at most z users. For each edge in M_z we say that the adjacent user and server are *matched*. Thus, one server can be matched to at most z users. The *capacity* C of the matching is defined as the number $|M_z|$ of edges in the matching, i.e. its cardinality. In accordance with previous work [41], the *energy* H is defined as

$$H = zS + N - 2C. \quad (3)$$

Hence, H is equal to the number of unmatched users plus the number of users that all servers can still host. A low value of the energy means that the given resources are well used, such that many, or all users are served

and at the same time not too many, possibly no, servers exhibit unused serving capacities.

For the maximum z -matching problem one wants to find a z -matching M_z which maximises the capacity C . This will usually depend on the edges in the graph. For any graph, the capacity C is bounded by a theoretical capacity which is obtained, when all users are matched to servers or when all servers are matched to z users. This leads to the maximum theoretical capacity

$$C_{\max} = \min(N, zS) = \min\{1, z/\eta\}N. \quad (4)$$

Note that C_{\max} does not at all depend on the edges, i.e. the actual topology of the graphs. But it depends on the given graph G whether this maximum can actually be reached. Naturally, the more edges exist, the more likely it is that a given graph reaches C_{\max} . For numerical reasons, we do not require that C_{\max} is 100% reached. Instead, we call a graph *saturated* if the actual capacity reaches γC_{\max} with $\gamma \leq 1$ being a suitable threshold. Correspondingly, when studying an ensemble of random graphs, the saturation probability is defined as

$$p_{\text{sat}} = \text{Prob}(C \geq \gamma C_{\max}). \quad (5)$$

We use p_{sat} as an order parameter. If p_{sat} is close to zero, few graphs reach the capacity γC_{\max} . We call this the unsaturated phase. But if p_{sat} is close to one, almost all graphs have a high capacity. This describes the saturated phase. As our results will show below, we are indeed able to observe phase transitions between these two phases.

In principle, one could use $\gamma = 1$, but for large random graphs, for most ensembles, it can be anticipated that it is exponentially unlikely that all demands can be matched exactly to all resources. We confirmed this in our numerical experiments. Thus, we used a value of γ close to 1, i.e. $\gamma = 0.9$, hence we call a network saturated if there is an almost complete balance between demand and resources. We also verified by tests that our results did not change significantly when we used other values such as $\gamma = 0.95$ or $\gamma = 0.85$.

Note that the variance of p_{sat} is given simply by $\sigma_{p_{\text{sat}}}^2 = p_{\text{sat}}(1 - p_{\text{sat}})$. We used it to calculate our error bars and to obtain more conveniently the positions of the phase transitions; see Sect. 4.3.

2.2 Random graphs

We consider the same two different networks ensembles as previously studied [41]. The first networks ensemble consist of (k, q) -regular graphs where all users have the same fixed integer degree k and all servers have integer degree q , satisfying Eq. 2. To generate such random graphs for the numerical studies, the configuration model [42, 43] can be used. In our case for (k, q) -regular graphs, we did the following: k stubs are assigned to each user and q stubs to each server. Then, iteratively one stub from a user and one stub from a server are drawn randomly (with uniform distribution from the

list of free stubs), respectively. If there is so far no edge between the two nodes where the stubs belong to, the edge connecting them is created and the used stubs get removed. If this edge already exists, two new stubs are drawn. This procedure is repeated until no stub is left. According to [44], redrawing two stubs if the edge already exists creates a bias in the generation of the random graphs. This bias can be removed, after the initial graph construction has finished, by repeatedly swapping the edges of the generated graph [45]. For a swap, two edges are chosen randomly. Then, the users, but not the servers to which the edges are connected are swapped. Repeating this ‘shuffle’ enough times will lead to unbiased graphs. We tested the influences of edge swaps and in our case, they had no measurable effect on the final data. This might be because we consider bipartite graphs. Thus, to save computation time, we did not perform such swaps.

The second network ensemble we considered, which is in the centre of our study, consist of bipartite Erdős–Rényi graphs [46]. This means, each possible edge between a user and a server is drawn with a probability p , with $p = k/S$ for any desired average number k of neighbouring servers for the users. Thus, the degree distributions of users and servers are Poissonian, respectively.

3 Methods

To solve the maximum z -matching problem, which we call also just z -matching in the following, numerically on a given graph, we map the problem to the original 1-to-1-matching problem on bipartite graphs. To achieve this, each server node is cloned $z - 1$ times. This means that $z - 1$ new nodes will be inserted in the graph for every server node. Then, for each user in the neighbourhood of the server, an edge to each of the new $z - 1$ nodes is created. Thus, in the graph with cloned nodes, each clone has the same neighbours as the original server node. Next, a matching is calculated for the graph with the cloned nodes. Here, each node is matched at most once. This means, with respect to the original graph, each user node will be matched at most once and each server node will be matched at most z times. Therefore, a standard 1-to-1-matching on the graph with cloned nodes corresponds to a z -matching on the original graph. This procedure is shown in Fig. 1.

To find a maximum matching, we used Edmonds’ Blossom Shrinking algorithm [47], implemented in the LEMON-library [48].

4 Results

We have performed simulations [49] using exact numerical matching calculations for the two graph ensembles, for various values of the parameters, for various

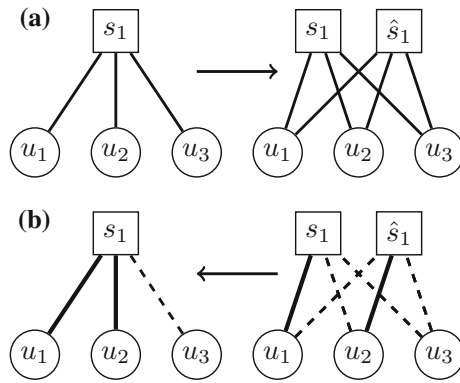


Fig. 1 **a** Cloning the server s_1 for the case $z = 2$ to create \hat{s}_1 having the same neighbours as s_1 . **b** A matching on the graph with cloned servers corresponds to a z -matching on the original graph. Edges in the matching are marked bold, non-matched edges are marked dashed

graph sizes of up to $N = 500000$ user nodes and up to $S = 250000$ server nodes. We performed for all results an average over up to several thousands of different graph realisations. Details are stated below. We first compare our numerical results with the previously obtained analytical results [41]. Next, we investigate the finite-size scaling behaviour of the capacity and compare with scaling form previously found [26] for standard matchings. In the main part, we show the results concerning the saturable–unsaturable phase transition and compare with the algorithmic behaviour.

4.1 Comparison with previous results

We studied the (k, q) -regular graphs, i.e. for integer values of k . We considered a rather large graph size $N = 40000$ and $\eta = 4$, thus $S = 10000$. We have considered $k = 3$, thus $q = \eta k = 12$. In Fig. 2, the average capacity per user node $\langle C/N \rangle$ and the average energy per user node $\langle H/N \rangle$ are shown as a function of z . The results are averaged over 100 realisations for each value of z . But due to the simple structure of the graphs, although being random, there are no statistical fluctuations on the results.

One can observe three cases, dependent on the value of z :

- For $z < \eta$, or equivalently $zk < q$, the capacity is always $C = zS = C_{\max}$ and the energy is $H = N - zS > 0$. All servers are matched to z users, but there are more users than all servers together can handle.
- At $z = \eta$, i.e. $zk = q$, the capacity is $C = N = C_{\max}$ and the energy is $H = 0$. This is an optimal situation since all users are matched to servers and no server has unused resources.
- For $z > \eta$, i.e. $zk > q$, the capacity is $C = N = C_{\max}$ and the energy is $H = -N + zS > 0$. All users are matched, but there are more servers than needed.

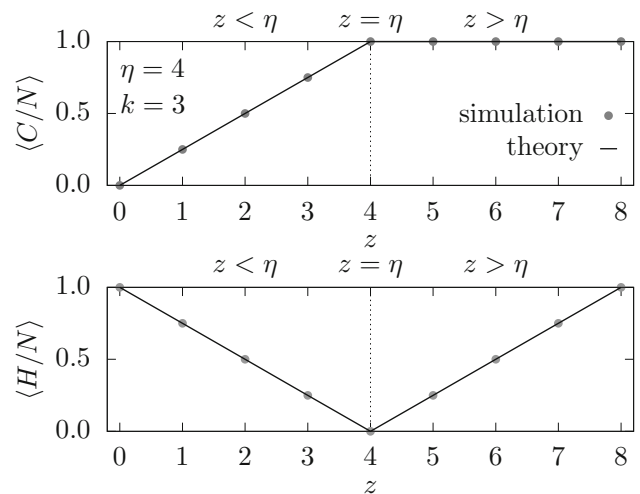


Fig. 2 Capacity density C/N and energy density H/N of the z -matching as a function of z for (k, q) -regular graphs with $N = 40000$ user nodes. The circles denote the results of the simulation and the line represents the analytical predictions from [41]. The lines are actually guide to the eyes which connect the predictions which are available only for integer values of z as well. The dashed vertical line marks $z = \eta$. There are no error bars because the results are always identical

Thus, in all networks, the full capacity is reached, but there is only one point of optimal balance between user demands and provided resources, where the energy is zero. It is interesting, that such a point of balance is possible. Note that here with $k = 3$, the graphs exhibit a rather large number of edges much larger than the number of edges in the matching. On the other hand for the smallest meaningful degree $k = 1$, at $z = \eta$, all users will be connected to exactly one server, and, because of $q = \eta k = z$, each server will be connected to z distinct nodes. Thus, trivially the full edge set is a z -matching and again all user demands are satisfied. Anyway, these results confirm those recently obtained analytically [41] for this network ensemble.

More variations in the results are obtained for the other network ensemble we have studied, the bipartite Erdős–Rényi graphs, since for this ensemble the nodes exhibit fluctuations of the degrees. In Fig. 3, the average capacity $\langle C/N \rangle$ per user node is shown as a function of the average user degree k for server capacity $z = 2$ and $N = 500000$ users. The results are averaged over 500 realisations for each value of k . Two cases for the user to server ratio are considered, $\eta = 2$ and $\eta = 4$. These values correspond to average server degrees $q = 2k$ and $q = 4k$, respectively, and server numbers $S = N/\eta = 250000$, and $S = 125000$, respectively. When approaching large degrees k , in both cases the limiting capacities are reached which are according to Eq. 4 $C_{\max}/N = 1$ for $\eta = 2$ and $C_{\max}/N = 0.5$ for $\eta = 4$. Our results from using the exact algorithm agree well with the previous results [41] obtained by a message-passing algorithm. Note that we obtained data

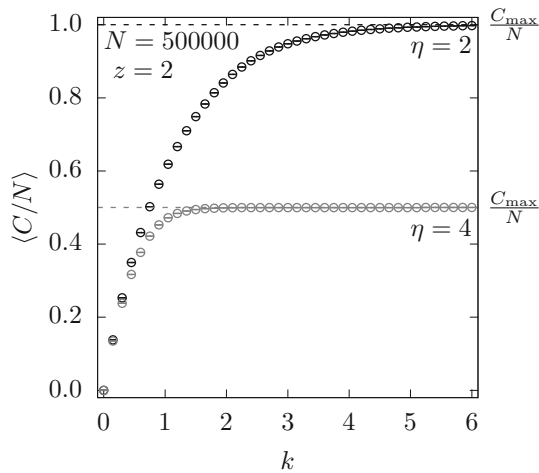


Fig. 3 Average capacity density $\langle C/N \rangle$ as a function of the mean user degree k for bipartite Erdős–Rényi graphs, $N = 500000$, $z = 2$, for two cases $\eta = 2$ and $\eta = 4$. The error bars are smaller than the point size

for all values of q down to zero, while in the previous work only the range $q > z$, i.e. $k > z/\eta$ was considered. Anyway, this success of the message-passing approach in the previous work is interesting, because it is known that for other problems, like the NP-hard vertex-cover problem, message-passing fails in the range of high degrees because for such models there exist replica-symmetry breaking [14, 15, 17]. The reason for the success with respect to z -matching could be that in the range of large values of k , the problem is easy to solve since there are enough options for each user and each server, which do not block each other too much. This corresponds to a simple, i.e. “dense” organisation of the solution space and makes a quick convergence of the message-passing iterations possible. On the other hand for NP-hard problems, each assignment of a problem variable has typically a strong impact on the availability of suitable assignments for other variables.

4.2 Finite-size behaviour

Since there are basically no finite-size effects for the (k, q) -regular graphs, we studied the finite-size scaling behaviour for bipartite Erdős–Rényi graphs. For $N \leq 1000$, we averaged over 5000 different realisations and for larger values of N , a number of 500 realisations turned out to be sufficient.

In Fig. 4, the average capacity density $\langle C/N \rangle$ is shown as a function of the system size for the case $k = 2$, $\eta = 2$ and $z = 2$. Motivated by the results [26] for standard matching, we fitted the data to the function

$$\langle C/N \rangle = c_\infty + \alpha_1 \frac{1}{N} \tag{6}$$

and found a good agreement. The resulting parameters, also for the other cases we have studied, are shown in Table 1. For $\eta = 4$, $z = 2$ and $k = 3$, the fit does not converge since most graphs reached C_{\max} , and there-

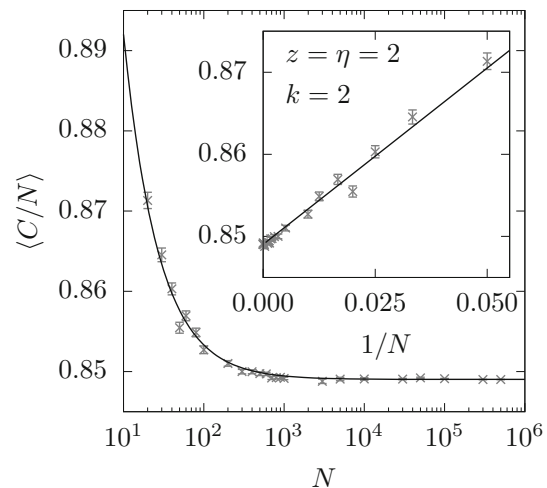


Fig. 4 The average capacity density $\langle C/N \rangle$ as a function of the number of users N for bipartite Erdős–Rényi graphs. The line represents the fit according to Eq. 6. Inset shows $\langle C/N \rangle$ as function of $1/N$ for the same data to confirm the $1/N$ -behaviour

Table 1 Maximum capacity C_{\max} and fit parameters for the scaling behaviour Eq. 6 for different values of the parameters η , z and k

η	z	k	C_{\max}/N	c_∞	α_1
2	2	1	1	0.60089 (2)	0.32 (1)
		2	1	0.84902 (2)	0.43 (1)
		3	1	0.947900 (9)	0.388 (6)
4	2	1	0.5	0.466631 (9)	0.118 (8)
		2	0.5	0.499142 (1)	0.0141 (6)
		3	0.5	–	–
4	4	1	1	0.62561 (2)	0.70 (2)
		2	1	0.86292 (2)	1.09 (1)
		3	1	0.950115 (9)	0.794 (6)

fore, a finite-size dependence is hardly to observe. For all other cases, we observe a good agreement with the $1/N$ scaling. Only the prefactor of this term seems to be non-universal. In addition, it should be noted that not only the finite-size capacities but also the limiting values of the capacity density are usually well below C_{\max}/N , due to typically too small average degrees. This corresponds also to the behaviour seen in Fig. 3.

4.3 Phase transition

Finally, we study the saturation probability p_{sat} . Due to the simple structure of (k, q) -regular graphs, we only focus on the Erdős–Rényi graphs and investigate them while varying the average degrees k and $q = \eta k$. All results are obtained over 500 difference realisations for each value of k and each system size $N, S = N/\eta$.

Figure 5 shows $p_{\text{sat}}(k)$ for different number N of users. p_{sat} increases from 0 to 1 with growing degree k , such that the curves become steeper when increas-

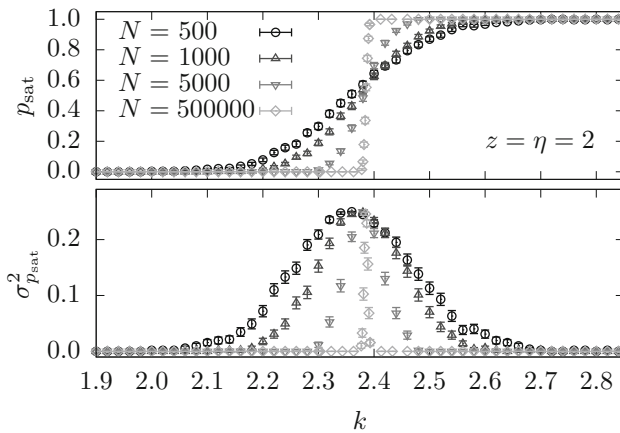


Fig. 5 Saturation probability p_{sat} (top) and its variance $\sigma^2_{p_{\text{sat}}}$ (bottom) as a function of k for different number N of users. For the largest number of users, there is an almost step-wise transition from $p_{\text{sat}} = 0$ to $p_{\text{sat}} = 1$ at some value of $k = k_c$

ing N . This is an indication for a transition from an unsaturated phase to a saturated phase. Note that this transition becomes almost step-wise for a really large number of users. We are able to observe the phase transition in such a clear way, because we could study huge system sizes due to the polynomial nature of the problem. This is in contrast to previously studied phase transitions for NP-hard optimisation problems, where only exponential-time exact algorithms are known, and therefore, only rather small system sizes could be studied exactly.

To study the observed phase transition in more detail, we use finite-size scaling analysis [50]. Hence, we assume that for continuous transitions the saturation probability follows the standard finite-size scaling relation

$$p_{\text{sat}}(k, N) = \tilde{p}_{\text{sat}}((k - k_c)N^{1/\nu}), \tag{7}$$

with infinite-size critical point k_c and ν being the exponent describing the divergence of the correlation length. We obtained the best-fitting scaling parameters using the tool `autoScale.py` [51]. The results are obtained over 9 different system sizes N , ranging from 500 to 500000. Figure 6 shows the resulting data collapse for $\eta = 2, z = 2$. Apparently, the collapse works very good for this case. The best found values for k_c and ν for all studied cases can be found in Table 2. The quality \mathcal{S} is the average deviation of the data points from the collapse curve, measured in terms of error bars [52]. Since \mathcal{S} is close to one for all cases we have considered, the quality of the data collapse is always very good.

The statistical errors of the scaling parameters are determined as how much a parameter has to be changed to increase the quality \mathcal{S} of the fit by one to $\mathcal{S} + 1$. We also systematically changed the intervals over which the data collapse is performed. However, these differences turned out to be smaller than the statistical errors, so

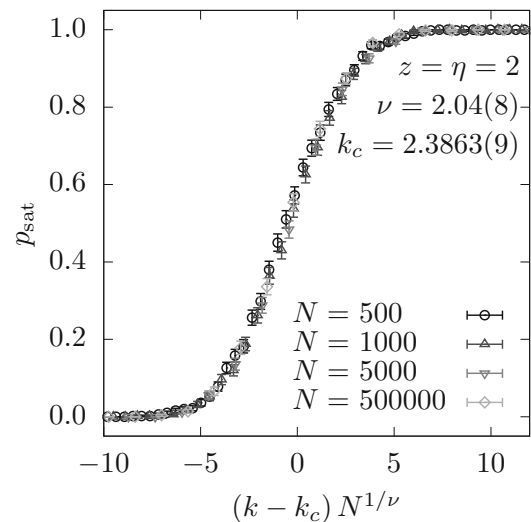


Fig. 6 Finite-size scaling for the saturation probability. Within the statistical fluctuations, the values for different N overlap. For visualisation, only four system sizes N are shown, but the results were obtained including all measured sizes N

Table 2 Results for the finite-size scaling parameters k_c, ν and the quality \mathcal{S} of the fit, for the considered cases of η and z

η	z	k_c	ν	\mathcal{S}
2	2	2.3863 (9)	2.04 (8)	0.90
4	2	0.8852 (4)	2.0 (1)	0.84
4	4	2.312 (1)	2.0 (1)	1.48

we state only those. Interestingly, within error bars, the value for ν is compatible with a value of $\nu = 2$ in all studied cases. This indicates that the behaviour of the saturable–unsaturable phase transition is universal with respect to network parameters.

Note that the phase transition can also be studied and analysed by obtaining the variance $\sigma^2_{p_{\text{sat}}}$ as a function of k . Although $\sigma^2_{p_{\text{sat}}}$ does not contain any additional information, it is easy to analyse, because it peaks at the apparent transition for each system size, which can be seen in the bottom of Fig. 5. To confirm the results obtained from p_{sat} , we also analysed the phase transition by a finite-size analysis of the variance. In the thermodynamic limit, the variance should be maximal at the critical point. On finite systems, the position of maximal variance, denoted as k_{max} , will approach k_c as the system size grows. To determine k_{max} more accurately than given by the resolution of the considered values of k , we performed Gaussian fits to $\sigma^2_{p_{\text{sat}}}(k)$ in small intervals near the maxima. An example is shown in the inset of Fig. 7. To extrapolate k_{max} the fit,

$$k_{\text{max}}(N) = k_c + \beta_1 N^{-1/\nu} \left(1 + \beta_2 N^{-1/\nu} \right) \tag{8}$$

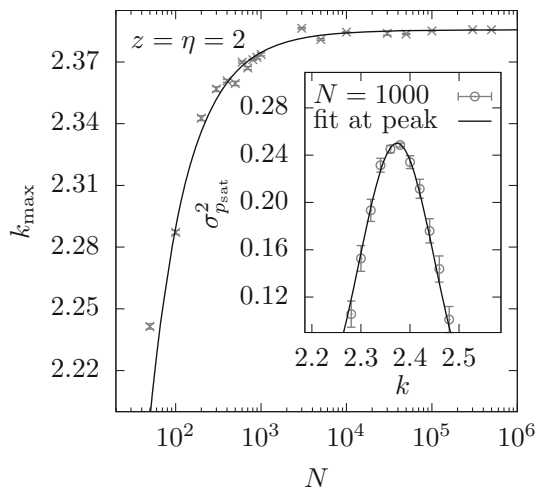


Fig. 7 Position of maximal variance k_{\max} as a function of the number of users N . The line represents the fit. The inset shows an example for the Gaussian fit around the peaks of the variance $\sigma_{p_{\text{sat}}}^2$ of the saturation probability. Note that the peak position shifts to the right for larger values of N

Table 3 Parameters k_c , β_1 and β_2 obtained from the fit of the position of maximum variance $k_{\max}(N)$ according to Eq. 8

η	z	k_c	β_1	β_2
2	2	2.38566 (3)	-0.07 (1)	137 (21)
4	2	0.88487 (2)	0.076 (6)	-68 (5)
4	4	2.31068 (4)	-0.04 (1)	404 (118)

is used. Note that we had to use here a correction term to the scaling behaviour, taking care of the very small system sizes. But we did not need to add a correction exponent to achieve a good fit and used exponent $2/\nu$ instead. In addition, based on the above results of the finite-size scaling, we fixed $\nu = 2$. Figure 7 shows the fit for the case $\eta = z = 2$. The results of the obtained fit parameter for all three cases are shown in Table 3. Within error bars, one or two sigma, the values for k_c agree with the above results which we obtained from the data collapse.

To summarise, our results speak in favour of a phase transition from a unsaturable to a saturable phase at a critical average degree k_c which depends on the graph structure. The scaling of the phase transition seems to be governed by a universal exponent $\nu \approx 2$, which is very different from the usual mean-field exponent $\nu = 1/2$.

4.4 Algorithm running time

We next analyse the running time of the matching algorithm when varying the average user degrees, to see whether the phase transition is reflected for this quantity. Unfortunately, the package we used, the fastest open-source matching algorithm implementation to our

knowledge, does not provide a machine-independent measure of the running time. Therefore, we had to limit our self to measure the CPU time. For this purpose, we used always the same machine under the same conditions. Figure 8 shows the median of the CPU time for Erdős–Rényi graphs as the average degree increases. Interestingly, the CPU time increases rapidly around the critical point, a typical behaviour observed so far for a phase transitions in NP-hard optimisation problems.

Note that there appears to be a small non-monotonicity at $k = 2$. Interestingly, the LEMON implementation of the matching algorithm changes the behaviour at average degree 4. For graphs with less number of neighbours, first a greedy heuristics to create an initial matching is applied, followed by the standard Edmonds’ algorithm. For denser graphs, no greedy initialisation is performed but a special variant of Edmonds’ algorithm, postponing so called “shrinks”, is used. Here, for $k = 2$, each user has on average two servers available in the original graph. Since the case $z = 2$ is considered, it means that, through the duplication of the server nodes, for the cloned graph each user has degree $zk = 4$. The degree of the server nodes is $q = \eta k = 4$ as well. Thus, at $k = 2$, the overall degree is 4 for the graph actually supplied to the matching code, which leads to a change of the algorithmic details and explains the jump in the running time.

Beyond this direct effect, there are other cases of matching problems, where certain degree values play a crucial role for the properties of matchings in networks. For example, the *control problem* of a dynamical system can be mapped to a maximum-matching problem [53]. Here, it was found that sudden changes of the fraction of matched nodes, which are those to control in the original problem, take place when the fraction of nodes exhibiting small degrees 0,1,2 reach certain thresholds [54]. This will likely also lead to strong changes of the performance of various matching algorithms as function of the degrees.

For the present problem, for much larger values k , the median CPU time decreases slowly, even though there are more edges to handle. This is the case, because the z -matchings become more and more degenerate in this region, i.e. the algorithm has more feasible options to choose among.

4.5 Approximation algorithm

The result of the running time of the exact algorithm shows that for small values of k , matchings can be obtained quickly. This could mean that in that region they are so simple to obtain such that even a non-exact but even faster algorithm is feasible. Therefore, we compare the exact matching algorithm with a matching heuristics. We considered the commonly used so called minimum-degree heuristics [56]. It finds a matching by connecting the nodes with lowest degree first, until no more nodes can be matched. The basic idea is that for nodes with few neighbours, one has to find a matching

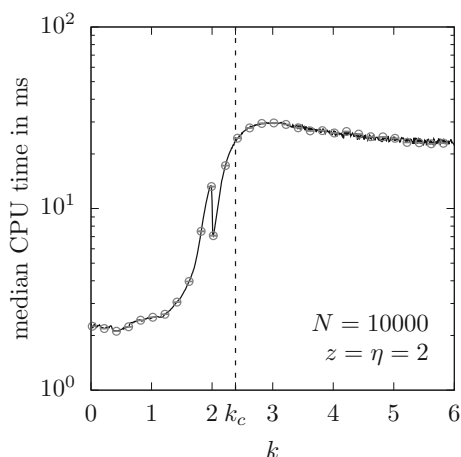


Fig. 8 The median of the CPU time as function of user degree k , each value averaged over 1500 runs. The error bars are obtained using bootstrap resampling [55]. The circles show only some data points for better visualisation, but the black line connects all measured points. Note the bump at $k = 2$ which is probably related to a fast heuristics used by the algorithm. The vertical dashed line marks the critical point k_c as obtained above

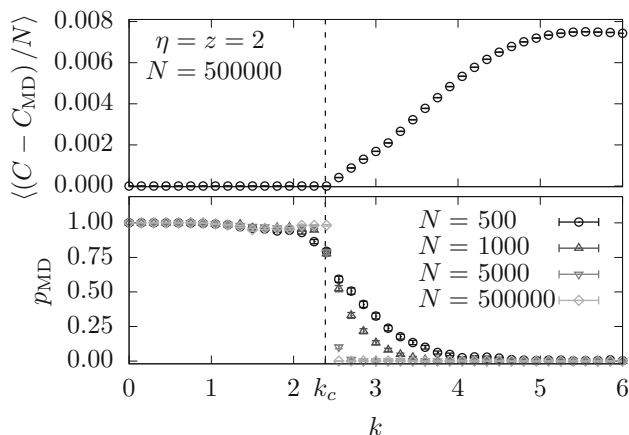


Fig. 9 Top: the difference of the capacity densities C/N obtained from the exact matching algorithm and from the minimum degree (MD) heuristics on the same graphs. Bottom: p_{MD} as a function of k for several system sizes N . The vertical dashed line marks the critical point k_c

partner first, while nodes with many neighbours will still find a partner even if many of their neighbours have been matched already. Note that the heuristics has a linear running time. In this context, we study p_{MD} , i.e. the empirical estimated probability that the minimum-degree heuristics obtains the same capacity as the exact matching algorithm. The top of Fig. 9 shows how far the capacity density C/N obtained from the minimum-degree heuristic differs from the one calculated by the exact matching algorithm. On the bottom, $p_{MD}(k)$ is shown. Both results are obtained over 500 different realisations for each value of k . Note that the average difference of the capacities between both algorithms

starts to grow for $k \geq k_c$. In particular, p_{MD} undergoes for large systems an almost step-wise transition from 1 to 0 near the critical point. This means that for $k < k_c$ the heuristic finds solutions with are well comparable to the exact solution. But for $k > k_c$ this is no longer the case. Hence, the saturable–unsaturable phase transition coincides with a kind of easy–hard transition with respect to a fast heuristics, although the z -matching problem is polynomially solvable everywhere.

5 Summary and outlook

We have studied the saturable–unsaturable phase transition for the z -matching problem on bipartite Erdős–Rényi random graphs. Since the problem can be solved with exact algorithms in polynomial time, we could study very large systems with good accuracy, leading to high-precision estimates of the critical points k_c and of the critical exponent ν of the correlation length.

We have also studied the running time of the exact algorithm and found that the phase transition point is very close to the largest change in the running time. In addition, for the minimum-degree heuristics, when studied for increasing node degrees, we find that the degree beyond which the heuristics start to fail, seems to agree with the critical point k_c . Thus, the saturable–unsaturable transitions seems to coincide with strong changes of the algorithmic behaviour. This was previously observed mainly for NP-hard optimisation or constraints-satisfaction problems, not for polynomial problems.

Thus, for future work, it could be very interesting to study other polynomial optimisation problems in a similar way and verify whether phase transitions in connection with changes of the run time are present. This could also apply to the investigation of other ensembles of the z -matching problems, or other variants of matching. This could lead to better understand the relation between phase transitions and computational hardness of optimisation or decision problems.

Finally, to understand this relation even better, one could analyse the solution structure for the z -matching problem. For this purpose one could extend the algorithm to allow for sampling of degenerate solutions, possibly by introducing slight random changes to the initially uniform edge weights, to make so-far found solutions unfavourable. This would allow to investigate the cluster structure [57] of the solution space. Possibly one could in this way observe changes of the solution-space structure in connection with the saturable–unsaturable transition, as it has been the case for NP-hard problems [58–60].

The simulations were performed at the HPC cluster CARL, located at the University of Oldenburg (Germany) and funded by the DFG through its Major Research Instrumentation Program (INST 184/157-1 FUGG) and the Ministry of Science and Culture (MWK) of the Lower Saxony State.

Author contributions

A.K. Hartmann and M. Fränzle designed the study. T. Kahlke coded the programs, performed the simulations, and analysed the results. A.K. Hartmann and T. Kahlke wrote the manuscript. All the authors confirmed the final manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data availability statement This manuscript has no associated data or the data will not be deposited. [Authors' comment: No external data, like from experiments, was collected in this study. All results are obtained directly from simulations and can be seen as they are in the figures.]

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- O.C. Martin, R. Monasson, R. Zecchina, *Theor. Comput. Sci.* **265**(1), 3 (2001). [https://doi.org/10.1016/S0304-3975\(01\)00149-9](https://doi.org/10.1016/S0304-3975(01)00149-9). Phase Transitions in Combinatorial Problems
- A.K. Hartmann, M. Weigt, *Phase Transitions in Combinatorial Optimization Problems* (Wiley-VCH, Weinheim, 2005)
- C. Moore, S. Mertens, *The Nature of Computation* (Oxford University Press, Oxford, 2011)
- M. Mézard, A. Montanari, *Information, Physics and Computation* (Oxford University Press, Oxford, 2009)
- M.R. Garey, D.S. Johnson, *Computers and Intractability* (W.H. Freeman, San Francisco, 1979)
- R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, L. Troyansky, *Nature* **400**, 133 (1999)
- S. Kirkpatrick, B. Selman, *Science* **264**, 1297 (1994)
- S. Cocco, R. Monasson, *Phys. Rev. Lett.* **86**, 1654 (2001)
- R. Mulet, A. Pagnani, M. Weigt, R. Zecchina, *Phys. Rev. Lett.* **89**, 268701 (2002)
- M. Mézard, G. Parisi, *J. de Phys, France* **47**(8), 1285 (1986)
- I.P. Gent, T. Walsh, *Artif. Intell.* **88**(1–2), 349 (1996). [https://doi.org/10.1016/S0004-3702\(96\)00030-6](https://doi.org/10.1016/S0004-3702(96)00030-6)
- H. Schawe, A.K. Hartmann, *Europhys. Lett.* **113**, 30004 (2016). <https://doi.org/10.1209/0295-5075/113/30004>
- H. Schawe, J.K. Jha, A.K. Hartmann, *Phys. Rev. E* **100**, 032135 (2019). <https://doi.org/10.1103/PhysRevE.100.032135>
- M. Weigt, A.K. Hartmann, *Phys. Rev. Lett.* **84**, 6118 (2000)
- A.K. Hartmann, M. Weigt, *Theor. Comp. Sci.* **265**, 199 (2001)
- M. Weigt, A.K. Hartmann, *Phys. Rev. Lett.* **86**, 1658 (2001)
- M. Weigt, A.K. Hartmann, *Phys. Rev. E* **63**, 056127 (2001)
- S. Mertens, *Phys. Rev. Lett.* **81**, 4281 (1998)
- S. Mertens, in *Computational Complexity and Statistical Physics*, ed. by A.G. Percus, G. Istrate, C. Moore (Oxford University Press, New York, 2006), p. 125
- M. Mézard, G. Parisi, R. Zecchina, *Science* **297**, 812 (2002)
- A.K. Hartmann, H. Rieger, *Optimization Algorithms in Physics* (Wiley-VCH, Weinheim, 2001)
- A.K. Hartmann, H. Rieger (eds.), *New Optimization Algorithms in Physics* (Wiley-VCH, Weinheim, 2004)
- D. Jungnickel, *Graphs, Networks and Algorithms* (Springer, Heidelberg, 2010)
- M. Mézard, G. Parisi, *J. Phys. Lett.* **46**(17), 771 (1985). <https://doi.org/10.1051/jphyslet:019850046017077100.jpa-00232897>
- G. Parisi, M. Ratiéville, *Eur. Phys. J. B* **29**(3), 457 (2002). <https://doi.org/10.1140/epjb/e2002-00326-3>
- M. Mézard, G. Parisi, *J. de Phys, France* **48**(9), 1451 (1987)
- M. Mézard, G. Parisi, *J. de Phys, France* **49**(12), 2019 (1988). <https://doi.org/10.1051/jphys:0198800490120201900>
- J. Houdayer, J.H. Boutet de Monvel, O.C. Martin, *Eur. Phys. J. B* **6**(3), 383 (1998). <https://doi.org/10.1007/s100510050565>
- S. Caracciolo, C. Lucibello, G. Parisi, G. Sicuro, *Phys. Rev. E* **90**, 012118 (2014). <https://doi.org/10.1103/PhysRevE.90.012118>
- O.C. Martin, M. Mézard, O. Rivoire, *J. Stat. Mech: Theory Exp.* **2005**(09), P09006 (2005). <https://doi.org/10.1088/1742-5468/2005/09/p09006>
- R. Kenyon, ArXiv e-prints (2009)
- P. Kasteleyn, *Physica* **27**(12), 1209 (1961). [https://doi.org/10.1016/0031-8914\(61\)90063-5](https://doi.org/10.1016/0031-8914(61)90063-5)
- M.E. Fisher, *Phys. Rev.* **124**, 1664 (1961). <https://doi.org/10.1103/PhysRev.124.1664>
- H.N.V. Temperley, M.E. Fisher, *Philos. Mag.* **6**(68), 1061 (1961). <https://doi.org/10.1080/14786436108243366>
- M.E. Fisher, *J. Math. Phys.* **7**(10), 1776 (1966). <https://doi.org/10.1063/1.1704825>
- O.J. Heilmann, E.H. Lieb, *Commun. Math. Phys.* **25**(3), 190 (1972). <https://doi.org/10.1007/BF01877590>
- R.M. Karp, M. Sipser, in *Proc. 22nd Annual IEEE Symp. on Foundations of Computer Science*, vol. 22, ed. by I.C. Society (IEEE Computer Society, Los Alamitos, CA, USA, 1982), pp. 364–75
- L. Zdeborova, M. Mézard, *JSTAT* (2006). <https://doi.org/10.1088/1742-5468/2006/05/P05003>
- Y. Gu, W. Saad, M. Bennis, M. Debbah, Z. Han, *IEEE Commun. Mag.* **53**(5), 52 (2015). <https://doi.org/10.1109/MCOM.2015.7105641>

40. Z. Han, Y. Gu, W. Saad, *Matching Theory for Future Wireless Networks* (Springer, Heidelberg, 2017)
41. E. Kreačić, G. Bianconi, EPL (Europhysics Letters) **126**(2), 28001 (2019). <https://doi.org/10.1209/0295-5075/126/28001>
42. B. Bollobás, Eur. J. Combin. **1**(4), 311 (1980)
43. M.E. Newman, S.H. Strogatz, D.J. Watts, Phys. Rev. E **64**(2), 026118 (2001)
44. H. Klein-Hennig, A.K. Hartmann, Phys. Rev. E **85**, 026101 (2012). <https://doi.org/10.1103/PhysRevE.85.026101>
45. F. Viger, M. Latapy, Proceedings of the 11th Annual International Conference on Computing and Combinatorics **3595**, 440 (2005)
46. P. Erdős, A. Rényi, Publ. Math. Debrecen **6**, 290 (1959)
47. W.J. Cook, W.H. Cunningham, W.R. Pulleyblank, A. Schrijver, *Combinatorial Optimization* (Wiley, New York, 1998)
48. Egerváry Research Group on Combinatorial Optimization. LEMON stands for Library for Efficient Modeling and Optimization in Networks. It is a Open Sorce C++ Library. <http://lemon.cs.elte.hu/pub/doc/1.2.3/index.html>
49. A.K. Hartmann, *Big Practical Guide to Computer Simulations* (World Scientific, Singapore, 2015)
50. J. Cardy, *Finite-Size Scaling, Current Physics, Sources and Comments*, vol. 2 (North Holland, 2015)
51. O. Melchert, [arXiv:0910.5403v1](https://arxiv.org/abs/0910.5403v1) (2009). [arXiv:0910.5403](https://arxiv.org/abs/0910.5403)
52. J. Houdayer, A.K. Hartmann, Phys. Rev. B **70**, 014418 (2004). <https://doi.org/10.1103/PhysRevB.70.014418>
53. Y.Y. Liu, J.J. Slotine, A.L. Barabási, Nature **473**(7346), 167 (2011). <https://doi.org/10.1038/nature10011>
54. G. Menichetti, L. Dall'Asta, G. Bianconi, Phys. Rev. Lett. **113**, 078701 (2014). <https://doi.org/10.1103/PhysRevLett.113.078701>
55. B. Efron, R.J. Tibshirani, *An Introduction to the Bootstrap* (Chapman & Hall/CRC, Boca Raton, 1994)
56. I.S. Duff, K. Kaya, B. Uçcar. ACM Trans. Math. Softw. **38**(2) (2012). <https://doi.org/10.1145/2049673.2049677>
57. A.K. Jain, R.C. Dubes, *Algorithms for Clustering Data* (Prentice-Hall, Englewood Cliffs, USA, 1988)
58. W. Barthel, A.K. Hartmann, Phys. Rev. E **70**, 066120 (2004)
59. F. Krzakala, A. Montanari, F.R.T. rsenghi, G. Semerjian, L. Zdeborová, PNAS **104**, 10318 (2007). <https://doi.org/10.1073/pnas.0703685104>
60. A. Mann, A.K. Hartmann, Phys. Rev. E **82**, 056702 (2010)