



## INFORMATION, COMMUNICATION & COMPUTING

Fields of Expertise TU Graz

Source: istockphoto.com



**Kay Uwe Römer,**  
**Information, Communication & Computing**

Source: Lunghammer – TU Graz

**15** years after the Excellence Initiative was introduced in Germany, an Excellence program is now coming to Austria as well, with 36 letters of intent submitted over the summer out of which five will be funded initially. What did Austria learn from Germany? The German program originally foresaw graduate schools (PhD training), Excellence clusters (collaborative research), and future concepts

(an Excellence label for universities that managed to win a minimum number of Excellence clusters and graduate schools and proposed a convincing Excellence strategy). Both the graduate schools, (“there are other good funding instruments for PhD training”) and the future concepts (“Excellence should be assessed on past achievements, not on plans”) received substantial criticism, as a result of which the graduate schools were dropped, but the future concepts were continued largely unchanged. Given this assessment, it seems a wise choice that the Austrian version focuses on Excellence clusters initially. Other criticism of the German program included a division of universities into a few “Excellent” winners and many losers, and too much focus on research and too little on education. The total yearly funding of about 500m euros

of the whole program is ridiculous compared to Harvard’s annual multi-billion-dollar budget, and a huge waste of resources regarding writing proposal “bibles” of which only a tiny fraction is funded. Despite this, the German program has been successful without doubt in strengthening research and international visibility, especially by forming research teams of critical mass and having universities focus on their strengths. Let’s see how much of this carries over to Austria=Excellent and which new aspects we will experience over the coming decade.

In September Yannic Maus joined the Institute of Software Technology as the FoE ICC tenure-track assistant professor. In this edition of TU Graz research, he introduces himself and his research. Welcome, Yannic!

### Yannic Maus

## Foundations for Modern Distributed Computing

Many of today’s and tomorrow’s computer systems distribute data to several machines, or the systems are built on top of large-scale networks, such as the internet or sensor networks. Our research builds the theoretical foundations for these settings. We analyse and develop distributed algorithms that are fast and communication efficient.

Distributed algorithms are becoming increasingly important. Networks and the size of the data are often already too large to be stored, processed or controlled by a single central authority, or are distributed by design.

### BENCHMARK PROBLEM GRAPH COLORING

Let us illustrate this with an artificial example. Consider a large network of sensors deployed in the vineyards in the south



**Yannic Maus**  
**is an assistant professor at**  
**the Institute of Software Technology.**

Source: Yannic Maus

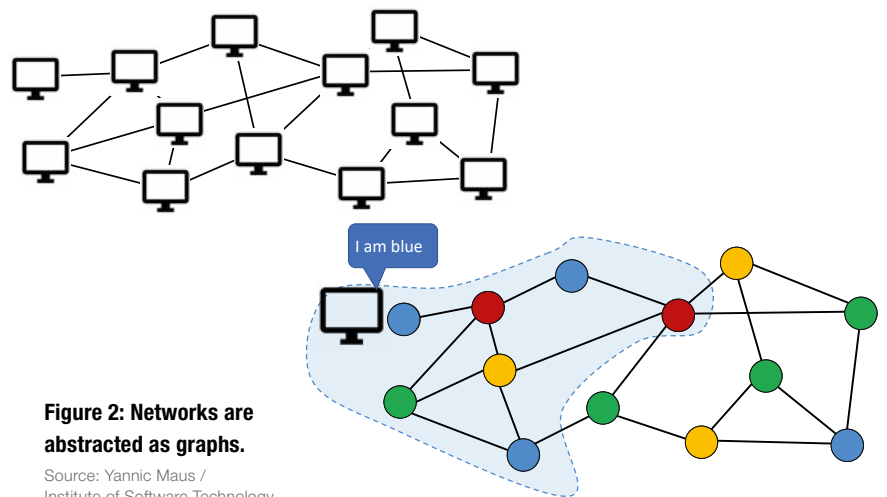




**Figure 1: Sensor networks are distributed if there is no central base station.**

Source: Yannic Maus /  
Institute of Software Technology

of Styria that measure parameters such as temperature or humidity. Due to the steep nature of the vineyards, obstacles and limited battery capacity, these sensors cannot communicate to a central base station. Instead, they build a multi-hop network and a sensor only communicates with the sensors in its vicinity. Furthermore, their communication is wireless, so two close-by sensors cannot use the same frequency to send their messages, as otherwise interference might occur. Thus, our goal is to assign frequencies such that close-by sensors do not receive the same frequency. This can be modelled as a classic graph coloring problem. The network is abstracted as a graph where each vertex resembles a sensor and each edge between two vertices indicates that the two sensors are so close that they cannot use the same frequency. The objective is to assign a colour (= frequency) to each vertex such that neighbouring vertices receive different colours. To save resources, one desires to use as few colours as possible. Although so simple, graph coloring has played an influential role in computer science, maths and also in distributed computing. Computing an assignment using the minimal number of colours cannot be solved efficiently unless  $P = NP$ . In the distributed setting, we use more colours,



**Figure 2: Networks are abstracted as graphs.**

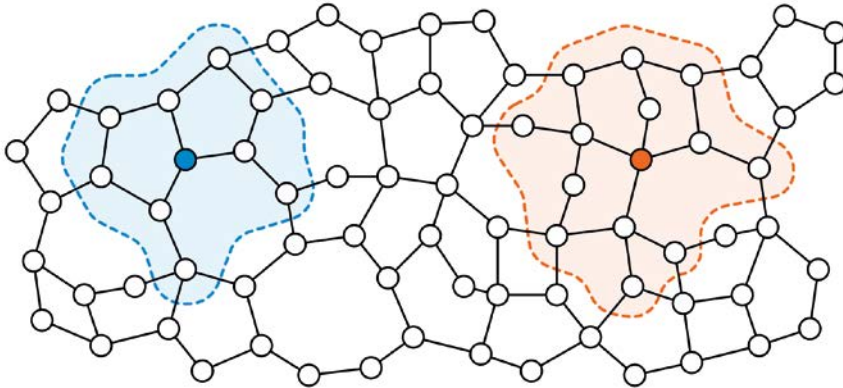
Source: Yannic Maus /  
Institute of Software Technology

but the problem still remains challenging. Recall our example. Each vertex of the graph is its own computer and does not know the network. How should one of these decide to use a specific colour without a central coordinator? Should it go for blue, red or green? It has to coordinate with its neighbours to not pick the same colour, and its neighbours have to coordinate with their neighbours, etc. Where do we begin? By rolling a dice.

### THE ROLE OF RANDOMNESS

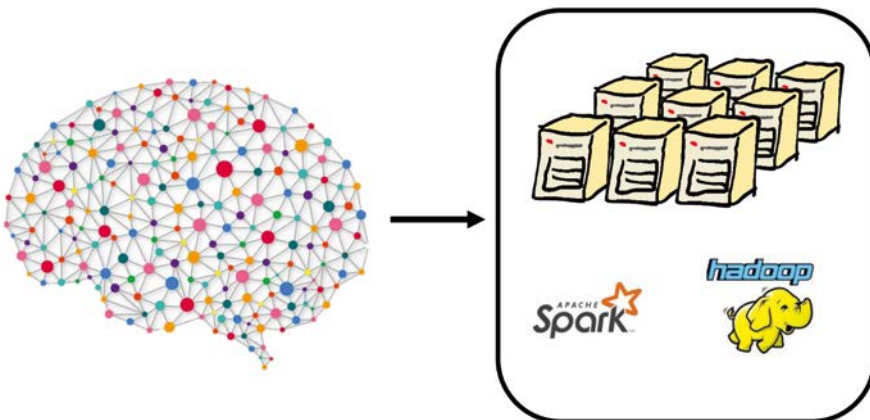
Let each node simultaneously pick a random candidate colour by rolling a dice. If no neighbour tries the same candidate

colour, stick to it, otherwise just roll the dice again. This is a simple and efficient algorithm. Unfortunately, its efficiency relies on having fair dices. Scientists call such algorithms randomized. Interestingly, many problems (not just the graph coloring problem) can be solved efficiently with randomization. Many of these have in common the fact that the correctness of a solution can be verified efficiently, e.g. you can ask your neighbour for its colour to see whether you have the same colour. But what if one wants so-called deterministic algorithms that do not use dices and are not only efficient if the “gods of the dices” are with us? Over the decades, >



**Figure 3: Classic distributed graph algorithms are allowed to use large messages and thus study the distance from which information in a graph has to be gathered to produce an output.**

Source: Yannic Maus /  
Institute of Software Technology



**Figure 4: The figure shows the massively parallel computing framework to which network algorithms contribute.**

Source: This figure is licensed under CC-BY.

the quest for efficient deterministic distributed algorithms has become the distributed version of the P vs. NP question: Can every problem whose solution can be verified efficiently by a distributed algorithm also be solved efficiently by a distributed algorithm? The classic P vs. NP question asks the same question for non-distributed algorithms. The answer to the distributed question is affirmative if we allow randomization. In order to also answer it for deterministic algorithms, we have built a large theory that relates randomized and deterministic complexities. Then, build-

ing on our theory, researchers from ETH Zurich have resolved the distributed P vs. NP question in the affirmative. Alas, these results come with expensive downsides; that is, they rely on a really large communication bandwidth in the network, something which is infeasible in practice.

### COMMUNICATION EFFICIENT AND SIMPLE ALGORITHMS

Thus, in recent years we have successfully focused on the design of communication- and bandwidth-efficient algorithms.

State-of-the-art algorithms are often very complicated and involved. Thus, besides being communication efficient and fast, one of our objectives is to design simple algorithms. Staying with our graph coloring example, we have shown that small changes to the aforementioned “roll-a-dice” algorithm suffice to make it exponentially faster.

### MASSIVELY PARALLEL COMPUTING

In large graph-processing frameworks, such as Google’s MapReduce or the open-source version Hadoop, a huge graph is arbitrarily distributed on a cluster of machines that communicate with each other in an all-to-all fashion. So, in contrast to the previous setting, there is no relation between the communication network and the input. Surprisingly, insights from network algorithms are very helpful in these frameworks, e.g. our communication-efficient algorithms are robust and run in such settings.

### ALGORITHMIC CHALLENGES IN OUR MODERN WORLD

Our future research will build foundations for the emerging algorithmic challenges in our modern world, e.g. the design of streaming and sublinear algorithms. Most real-world graphs are non-static in nature, e.g. the Facebook graph keeps changing when new users join or leave the network. We aim to design dynamic algorithms that react quickly to changes without recomputing solutions from scratch. The impact of distributed computing in these areas has been growing and new connections are being drawn each day.