What Can Be Computed Locally?

Michael Kaufmann

Distributed Computing Seminar, November 2003

1 Summary

The paper [1] deals with locality on several levels: It introduces a model for distributed computations and the notion of locally checkable labelings (LCLs). It proves properties about algorithms for locally checkable labeling problems. Two algorithms are analyzed in detail: One for the *Weak Coloring* problem and one for the *Formal Dining Philosophers* problem.

2 Locally checkable Labelings

The paper shows fundamental properties of local algorithms for LCL problems:

- *Undecidability:* In general, it's impossible to decide if a certain LCL problem has a local algorithm. However, it is possible to decide if the time bound is fixed.

Unfortunately, this proof doesn't give instructions on how to find an algorithm.

- *Randomization*: If there's a randomized local algorithm for a LCL problem, then a deterministic algorithm with the same time bound can be found.

This proof is important: We can concentrate on deterministic algorithms if we want to find one for a particular problem.

3 Weak Coloring

The *Weak Coloring* problem is quite artificial and has only few applications. It can be used where two types of resources are needed to perform an operation, but since every node only gets one type of resource, they have to cooperate.

3.1 Phase 1: Initial Coloring

The first phase creates a coloring with $d \cdot (d+1)^{d+2}$ colors. If $d \geq 8$, this expression is bigger than 2^{32}. In cases where we have less than 2^{32} IDs, we could take the ID as color number (example: IP addresses).

3.2 Phase 2: Assigning different Subsets

The authors don't give instructions on how to assign a different subset to every color number. Of course, in the model used (any computation on a single processor can be carried out in one time step) this isn't a problem, but in practice it is.

3.3 Phase 3: Runtime

The color reduction algorithm of the third phase is slower than that of the second phase. The paper states that it takes *O(number of colors)* rounds to decrease the colors to 2 in the worst case. But the runtime depends on the distribution of the colors in the graph. It may be interesting to study the average runtime of this algorithm, for example on random graphs.

3.4 Limitations

The paper only considers graphs where *d* (this is the maximal degree of a node in the graph) is known. Weak Coloring can be solved for graphs with unknown *d* (by replacing the algorithm of phase 2 with the algorithm of phase 3), but the runtime won't be constant anymore. It's an open question if a constant-time algorithm for graphs with unbounded *d* exists.

The second concern is the big number of colors that are generated in phase 1. This has been resolved by the follow-up paper [2] in which the authors describe a much better phase 1 algorithm that uses $d/2$ colors.

3.5 Example Graph: Equally colored Nodes

It was hard to find a sample graph where two nodes get the same color (Figure 1). Unfortunately the authors of the paper have not included any example graphs or figures that help to understand their paper.

4 Formal Dining Philosophers Problem

The paper uses the *Weak Coloring* problem to solve the *Formal Dining Philosophers Problem*. It's an extension of the Dining Philosophers Problem by Dijkstra: Instead of two specified forks, a processor can take any two forks that are in its reach.

4.1 Limitations

Often more than two processors need to share a resource. So this algorithm is suitable only for special applications.

5 Other local Algorithms

The Formal Dining Philosophers Problem is the only known non-trivial locally solvable LCL problem. Probably most LCL problems are trivial, and once we have a non-trivial instance, it's very hard to find an algorithm.

The follow-up paper [2] contains much improved algorithms compared to [1]. It also considers dynamic networks.

References

[1] Moni Naor and Larry Stockmeyer, *What Can Be Computed Locally?*, 25th Symposium on the Theory of Computing, 1993.

[2] Alain Mayer, Moni Naor and Larry Stockmeyer, *Local Computations on Static and Dynamic Graphs*, 3rd Israeli Symposium on the Theory of Computing and Systems, 1995.

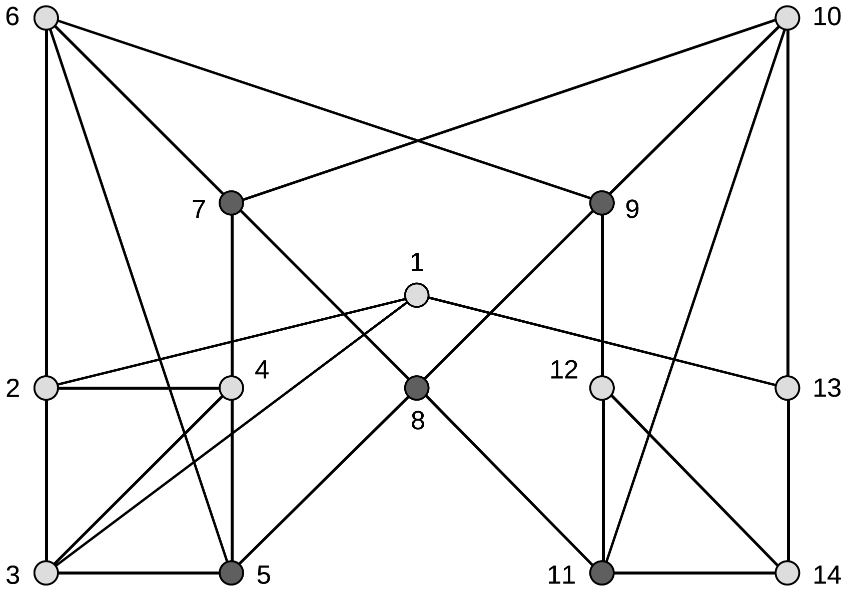

Figure 1: The nodes 5, 7, 8, 9 and 11 get the same color