# On Solving NP-complete Problems with Unconventional Models of Computation

Dominik Schultes

11. March 2004

**Abstract**

Concentrating on the algorithmic point of view, we summarize briefly two attempts of solving NP-complete problems in **polynomial time** with an unconventional model of computation:

- in [CPW+01] (described in [Bal01]), Chiu, Pezzoli, Wu, Stroock, and Whitesides solve instances of the *maximal clique problem* (MCP) using microfluidic networks, and
- in [Adl94] (described in [CPT01]), Adleman solves an instance of the *directed Hamiltonian path problem* using DNA Computing.

We want to emphasize one common problem of these attempts that shows that we need not only new models of computation, but also *new algorithms* in order to be able to solve big instances of NP-complete problems.

In their experiments, Chiu et al. show how to solve an instance of the maximal clique problem[1] in polynomial time. They obtain the speedup in comparison with a conventional implementation on a conventional machine by taking advantage of the possibility of parallelizing the following quite simple algorithm:

1. For each subset $V' \subseteq V$, set $\texttt{counter}[V'] = 0$.
2. For each edge $(u, v) \in E$, for each subset $V'$ with $u, v \in V'$, increment $\texttt{counter}[V']$.
3. For each subset $V'$, check if $\texttt{counter}[V'] = n'(n'-1)/2$, where $n' = |V'|$, i.e., check if there are enough edges so that the vertex induced subgraph $G' = (V', E')$ is a clique.
4. Return the largest clique.

The actual implementation bases on microfluidic networks, which can be constructed in parallel. In these networks the properties of fluids are used to search all potential solutions and pick out all solutions in parallel. In order to be able to count for each subset $V'$ the number of edges in the vertex induced subgraph $G' = (V', E')$, each edge $(u, v)$ leaves a tag at each subset it belongs to, i.e., at each subset $V'$ with $u, v \in V'$. This "leaving of a tag" is done by a flow of a fluid from an edge (represented by a reservoir) to each subgraph (represented by wells). The parallelism arises due to the fact that the fluid splits on its way from an edge to all corresponding subgraphs at each branching point and flows to all directions simultaneously.

In Adleman's experiment, an instance of the directed Hamiltonian path problem[2] is solved. The experiment bases on the following algorithm:

1. Generate random paths through the graph $G = (V, E)$.
2. Keep only those paths that
   - begin with the given source vertex $s$ and end with the given target vertex $t$,
   - have lengths $n = |V|$, and
   - contain all vertices of $G$ at least once.

[1] The problem is the determination of a clique of maximal size in a graph $G = (V, E)$, where a clique is a subgraph $G' = (V', E')$ of $G$ such that $\forall u, v \in V'$ with $u \neq v : (u, v) \in E'$, i.e., each pair of nodes is connected by an edge. This problem is known to be NP-complete.

[2] For a given graph $G = (V, E)$ and two given vertices $s, t \in V$ with $s \neq t$, the problem is the decision if there is a path from $s$ to $t$ that contains each vertex of $G$ exactly once. This problem is known to be NP-complete.

3. Return "Yes" if any paths remain, and "No" otherwise.

Similar to the first example, the speedup is obtained by parallelizing this algorithm. In order to do so, Adleman uses DNA computing by encoding vertices and edges in DNA in such a way that a large amount of random paths is formed in parallel (step 1). The filtering process (step 2) is done by operations on the DNA molecules, which are performed in parallel as well.

At first sight, both examples give the impression that we have now an efficient solution of big instances of NP-complete problems as only polynomial time is needed. But, it is very important to regard that the exponential complexity has *not* been eliminated, but it has only be shifted from time to space. In the first example, there is for each subgraph a well in the microfluidic network. There are $2^n - n - 1$ subgraphs that are considered (the $n + 1$ trivial subgraphs that contain only one node or are empty are ignored). Hence, the number of wells growths exponentially with the number of nodes $n$. In the second example, let us regard a complete graph as a worst case instance. There are $n!$ different simple paths of length $n$ (and, of course, there are even more paths if we count non-simple paths and paths of a length smaller than $n$ as well). If we want to use the given approach to compute a solution that is correct with a high probability, we have to consider at least an appreciable portion of these paths. As every considered path is generated and represented as DNA code, the required amount of DNA growths exponentially with the number of nodes $n$.

Of course, in both cases the authors are aware of this fact and the goal of their work is to show that such computations are possible on principle. For example, Adleman states "This experiment demonstrates the feasibility of carrying out computations at the molecular level." [Adl94] They do not claim that they can easily use their approach to solve big instances of the same problem.

However, let us have a look on the final goal of these attempts: The final goal should be to find solutions of big instances of NP-complete problems, i.e., we have to get rid off the exponential complexity both in time and in space. At the moment, there is no real improvement in sight – apart from perhaps better constants and a better energy efficiency, which Adleman mentions in [Adl94]. The fact that we now can solve instances of NP-complete problems in polynomial time seems to be an improvement, but we have already been able to do so without new models of computations: For example, in order to solve the MCP, we "just" take $2^n - n - 1$ processors and we assign one subgraph to each processor, which checks in polynomial time if the corresponding subgraph is a clique. Then, we iterate through all possible sizes in decreasing order starting with $n$. As soon as a processor represents a clique of the currently regarded size, it reports this clique as the maximal one – and we are done. So, in a sense, this approach is quite similar to the above mentioned experiments: We have a space-time tradeoff, but we have not eliminated the exponential complexity.

In conclusion, we want to emphasize that the experiments by Chiu et al. and by Adleman not only demonstrate that the presented new models of computation work on principle, but also show that it is probably not sufficient to find *new models of computation* unless appropriate *new algorithms* are found as well. In the regarded examples, the new models of computation were not able to break through the limits of the "old" algorithms, namely the exponential growth (either in time or in space).

# References


[Adl94] L. M. Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266:1021–1024, 1994.

[Bal01] Philip Ball. Liquid logic. nature science update, http://www.nature.com/nsu/010329/010329-8.html, March 2001.

[CPT01] C. S. Calude, G. Paun, and Monica Tatarm. A glimpse into natural computing. *J. Multi Valued Logic*, 7:1–28, 2001.

[CPW+01] D. T. Chiu, E. Pezzoli, H. Wu, A. D. Stroock, and G. M. Whitesides. Using three-dimensional microfluidic networks for solving computationally hard problems. In *Proceedings of the National Academy of Sciences USA*, volume 98, pages 2961–2966, 2001.