Chapter 12
Conclusions, Open Problems and Further Directions

We conclude with a number of open problems. Some of them are of a fundamental nature and some of them can serve as starting points for newcomers in the field.

Fundamental questions

Every problem in NP can be solved by enumerating all solution candidates. The question is whether such trivial enumeration can be avoided for every problem in NP. In other words, is brute-force search the only approach to solve NP problems in general? A positive answer to this question implies that $P \neq NP$. On the other hand, the assumption $P \neq NP$ does not yield a negative answer. Recent work of Williams demonstrates that: "... carrying out the seemingly modest program of finding slightly better algorithms for all search problems may be extremely difficult (if not impossible)" [219].

Most of the exact algorithms are problem dependent—almost every specific problem requires specific arguments to show that this problem can be solved faster than brute-force search. In the world of polynomial time algorithms and parameterized complexity we possess very powerful tools allowing us to establish efficient criteria to identify large classes of polynomial time solvable or fixed parameter tractable problems. It would be desirable to obtain generic tools allowing us to identify large classes of NP-complete problems solvable faster than by brute-force search.

Every algorithmic theory becomes fruitful when accompanied by complexity theory. In the current situation we are only able to distinguish between exponential and subexponential running times (subject to Exponential Time Hypotheses). A challenge here is to develop a theory of exponential lower bounds. For example, is it possible to prove (up to some plausible assumption from complexity like P$\neq$NP, FPT$\neq$W[1], ETH, etc.) that there is no algorithm solving 3-SAT on n variables in time 1.000000001^n?

More concrete questions

Three fundamental NP-complete problems, namely, SAT, TSP and GRAPH COLORING can be solved within the same running time $\mathcal{O}^*(2^n)$. Obtaining for any of

F.V. Fomin, D. Kratsch, *Exact Exponential Algorithms*, Texts in Theoretical Computer Science. An EATCS Series, DOI 10.1007/978-3-642-16533-7_12,

these problems an algorithm of running time $\mathcal{O}^*((2-\varepsilon)^n)$ for any $\varepsilon > 0$ would be exciting.

Can it be that for every $\varepsilon > 0$ the existence of an $\mathcal{O}^*((2-\varepsilon)^n)$ algorithm for one of these three problems yields an $O((2-\delta)^n)$ algorithm for the other two, for some $\delta > 0$? Recently, Björklund [22] announced a randomized algorithm solving HAMILTONIAN PATH in time $\mathcal{O}(1.66^n)$.

Many permutation and partition problems can be solved in time $\mathcal{O}^*(2^n)$ by dynamic programming which requires exponential space. An interesting question is whether there are $\mathcal{O}^*(2^n)$ time and polynomial space algorithms for TSP, GRAPH COLORING, and TREEWIDTH.

Some permutation problems like PATHWIDTH or TREEWIDTH can be solved in time $\mathcal{O}^*((2-\varepsilon)^n)$ (and exponential space). What can we say about DIRECTED FEEDBACK ARC SET, CUTWIDTH and HAMILTONIAN CYCLE?

The running time of current branching algorithms for MIS with more and more detailed analyses seems to converge somewhere near $\mathcal{O}^*(1.2^n)$. It appears that obtaining an algorithm running in time $\mathcal{O}^*(1.1^n)$ will require completely new ideas. Similarly the question can be asked whether MDS is solvable in time $\mathcal{O}(1.3^n)$. MINIMUM DIRECTED FEEDBACK VERTEX SET requires us to remove the minimum number of vertices of a directed graph such that the remaining graph is acyclic. The problem is trivially solvable in time $\mathcal{O}^*(2^n)$. The trivial algorithm was beaten by Razgon with an algorithm running in $\mathcal{O}(1.9977^n)$ time [178]. It seems that improving even to $\mathcal{O}^*(1.8^n)$ is a difficult problem.

SUBGRAPH ISOMORPHISM is trivially solvable in time $\mathcal{O}^*(2^{n \log n})$. Is it possible to solve this problem in time $2^{\mathcal{O}(n)}$? A similar question can be asked about GRAPH HOMOMORPHISM. In CHROMATIC INDEX (also known as EDGE COLORING) the task is to color edges with the minimum number of colors such that no two edges of the same color are incident. The only non-trivial algorithm we are aware of reduces the problem to (vertex) graph coloring of the line graph. This takes time $\mathcal{O}^*(2^m)$. Is CHROMATIC INDEX solvable in time $2^{\mathcal{O}(n)}$?

Enumerating the number of certain objects is a fundamental question in combinatorics. Sometimes such questions can be answered using exact algorithms. Consider the following general problem: "For a given property π, what is the maximum number of vertex subsets with property π in a graph on n vertices?" For example, the theorem of Moon-Moser says that when the property π is "being a maximal clique", then this number is $3^{n/3}$. But for many other natural properties, we still do not know precise (even asymptotically) bounds. For example, for minimal dominating sets the correct value is between 1.5704^n and 1.7159^n [88], for minimal feedback vertex sets between 1.5926^n and 1.7548^n [79]. For minimal feedback vertex sets in tournaments the old bounds of Moon [160]—1.4757^n and 1.7170^n—were recently improved by Gaspers and Mnich to 1.5448^n and 1.6740^n [102]. For minimal separators we know that the number is between 1.4423^n and 1.6181^n [95], for potential maximal cliques between 1.4423^n and 1.7347^n [96].