Algorithm Design Challenges
(Position Statement)

Giorgio Ausiello

Universita "La sapienza" Roma
Via Salaria 113, I-00198 Roma, Italy
ausiello@dis.uniroma1.it

Considering the tremendous growth of computer science in the last twenty five years, since the first personal computer made its appearance, and the even stronger impetus that to this growth was given by the creation of the World Wide Web, I do not think that anybody could dare to say what computing will be like during the 21st century. We are by now used to quite unexpected technological leaps that change the picture that is in our hands while we are still trying to analyze it and we do not know what else is waiting us behind the corner. May be, although I am quite skeptical, that some technological breakthrough will make molecular computers or quantum computers available earlier than we might expect and that this might dramatically change our view of computing.

On the other side, on a more conservative ground, if we look at a closer time scale, the next ten years for example, some driving trends of computing appear rather clearly. We may then be able to foresee the evolution of computing on the basis of the current directions of technology and applications and, accordingly, to pinpoint some of the contributions that theoretical research can give to such evolution.

First of all, the most impressive push to the technological evolution of computer science derives from the "all connected to all" paradigm and the use of IP as the basic support for this paradigm. This enormously increases the request for guaranteed quality of service performances on the internet in terms of voice and image communication, broadcasting, searching and web caching, correctness of mobile code migrations, security etc. Among other problems this development implies the study and design of new efficient and powerful techniques for search engines as it is witnessed by the Search Engine Watch and by the fact that important research institutions have started projects in this area such as the Clever Project (IBM, San Jose') and the Google Project (Stanford University).

An interesting aspect of the scientific development in this field is related to the fact that, in order to design web searching engines we need to represent and manipulate large portions of the web, a graph with millions of node and billions of arcs. This leads us to the second characteristic aspect of the current evolution of Information Technology: the need to manipulate very large input data. While software applications for the final user become (apparently at least) simpler and simpler, the applications involved with the management of ITC systems become more and more complex and resource demanding. For example, for defining their billing strategies telecommunication companies want to know the communication graph of all calls among their customers. Also scientific computer applications

J. van Leeuwen et al. (Eds.): IFIP TCS 2000, LNCS 1872, pp. 602–603, 2000.

require to deal with huge data sets (very long strings in biology, very large matrices in theoretical physics, very large instances for finite element methods in fluid dynamics etc). In all these cases the classical asymptotic analysis is not adequate for describing and interpreting the behaviour of programs. We have to rethink our computation models for allowing the analysis and design of algorithms on secondary storage and for predicting the impact of locality and cashing strategies on different algorithms. Also in this area, research efforts are already being carried out in various Universities worldwide, among others, Duke University in the US and the Universities participating in the European Union research project ALCOM-FT.

A third direction of computer evolution that can be clearly identified and that requires new answers from theoreticians is related to the increasing role of dynamic and on line applications. In this type of applications (think of routing problems, scheduling, resource allocation, investment strategies etc.) data are not static but evolve in a somewhat unpredictable way and we have to face such evolution by maintaining our data in such a way that we do not have to recompute from scratch the solution of a problem at any change of input data but we can just rearrange it with a limited cost. At the same time we want to define strategies for problem solution that, while not being optimal, anyhow are "competitive", that is do not lose too much with respect to an off line algorithm that, ahead of time, is aware of the whole sequence of changes in the input data. The efficient and competitive solution of dynamic and on line problems has been studied over the last fifteen years but still practical applications of theoretical results are limited by the fact that theoretical models are inadequate and a lot more research efforts should be spent in this domain.