Dagstuhl Seminar on

On-line Algorithms

June 24–28, 1996

Organized by:

Amos Fiat (Tel-Aviv University)
Gerhard Woeginger (TU Graz)

Summary

The Dagstuhl meeting on *On-line Algorithms* brought together 55 researchers with affiliations in Argentina (1), Austria (1), Canada (2), Czechia (1), Germany (4), Hungary (2), Israel (7), Italy (3), the Netherlands (5) and the USA (29). 34 presentations were given: There were 16 survey talks presented by

- Daniel Sleator (On the history of on-line algorithms)
- Avrim Blum (On-line algorithms in machine learning)
- Sandy Irani (Paging problems)
- Christos Papadimitriou (Non-standard models for competitive analysis)
- Susanne Albers and Jeffery Westbrook (Self-organizing data-structures)
- Serge Plotkin (On-line routing problems)
- Bala Kalyanasundaram and Kirk Pruhs (On-line problems in networks)
- David Johnson (On-line bin packing)
- Lawrence Larmore and Marek Chrobak (Metrical task systems and the k-server problem)
- Anna Karlin (Competitive analysis in practice)
- James Aspnes (Competitive analysis of distributed algorithms)
- Yossi Azar (On-line load balancing)
- Hal Kierstead (On-line graph coloring)
- Ran El-Yaniv (Decision-theoretic foundations of competitive analysis)
- Jiří Sgall (On-line scheduling)
- Yair Bartal (On-line distributed paging)

Moreover there were 18 shorter contributions that presented recent results. The abstracts of all these presentations are contained in this seminar report in alphabetical order. Moreover, there is a list of with some open problems that were mentioned in the open problem session.

The organizers enjoyed the social aspects of the meeting, especially the alcohol. We had a wine-and-cheese party every evening; on Tuesday evening the party included an open problem session organized by David Johnson. On Thursday evening we had wine-and-cheese along with a panel discussion chaired by Danny Sleator on "The Future of Competitive Analysis". Giorgio Ausiello and Alberto Marchetti-Spaccamela also discussed their plan to have yet another workshop on on-line algorithms in two years time, to be held in Sicily. Alan Borodin promised to come to Sicily. During this week, workshop participants consumed 74 bottles of wine, 163 bottles of beer, 13 bottles of mineral water, and 51 bottles of cola. On Wednesday evening, Kirk Pruhs organized an excursion to a pub in the nearby village where we watched the semi-finals of the European Soccer Championship.

Amos Fiat
Gerhard Woeginger

List of Participants

Susanne Albers, MPI Saarbrücken
James Aspnes, Yale University
Giorgio Ausiello, Universitá di Roma “La Sapienza”
Baruch Awerbuch, Johns Hopkins University
Yossi Azar, Tel-Aviv University
Amotz Bar-Noy, Tel Aviv University
Yair Bartal, ICSI Berkeley
Piotr Berman, Pennsylvania State University
Avrim Blum, Carnegie Mellon University
Allan Borodin, University of Toronto
Ran Canetti, Weizmann Institute
Marek Chrobak, University of California
Janos Csirik, Jozsef Attila University
Ran El-Yaniv, The Hebrew University
Ulrich Faigle, University of Twente
Anja Feldmann, AT& T Research
Esteban Feuerstein, University of Buenos Aires
Amos Fiat, Tel-Aviv University
Juan A. Garay, IBM T.J. Watson Research Center
Sandy Irani, University of California
David S. Johnson, AT& T Research
Bala Kalyanasundaram, University of Pittsburgh
Anna Karlin, University of Washington
Howard Karloff, Georgia Institute of Technology
Marek Karpinski, University of Bonn
Walter Kern, University of Twente
Hal Kierstead, Arizona State University
Rolf Klein, FernUniversitaet Hagen
Elias Koutsoupias, University of California at Los Angeles
Han La Poutré, University of Leiden
Lawrence L. Larmore, University of Nevada at Las Vegas
Stefano Leonardi, Universitá di Roma “La Sapienza”
Mark S. Manasse, Palo Alto
Alberto Marchetti-Spaccamela, Universitá di Roma “La Sapienza”
Ketsiya Meirman, The Hebrew University
Seffi Naor, Technion Haifa
Christos Papadimitriou, Berkeley
Steven Phillips, AT& T Research
Serge Plotkin, Stanford University

Kirk Pruhs, University of Pittsburgh
Adi Rosen, University of Toronto
Ronitt Rubinfeld, Cornell University
Baruch Schieber, IBM T.J. Watson Research Center
Sven Schuierer, Universität Freiburg
Jiří Sgall, Academy of Sciences of the Czech Republic
David Shmoys, Cornell University
Daniel Sleator, Carnegie Mellon University
Leen Stougie, University of Amsterdam
Eva Tardos, Cornell University
Andrew Tomkins, Carnegie Mellon University
Eric Torng, Michigan State University
Gyorgy Turan, The University of Illinois at Chicago
Zsolt Tuza, Hungarian Academy of Sciences
Arjen Vestjens, Eindhoven University of Technology
Joel Wein, Polytechnic University Brooklyn
Jeffery Westbrook, Yale University
Gerhard J. Woeginger, TU Graz

Abstracts

A Survey of Self-Organizing Data-Structures
Susanne Albers and Jeffery Westbrook

Self-organizing data structures modify their structure while processing a sequence of operations. The data structure is allowed to be in an arbitrary state, i.e. no global constraint has to be satisfied, but during each operation a simple restructuring rule is applied. The purpose of this restructuring is to guarantee the efficiency of future operations. Typically the restructuring is also done on-line, without knowledge of future operations. In this talk we review important results for self-organizing data structures such as linear lists and binary search trees.

The problem of maintaining self-organizing linear lists is also known as the list update problem. First we present results on the competitive ratio of deterministic on-line algorithms for the list update problem. We concentrate on the algorithms Move-To-Front, Transpose, Frequency-Count and Timestamp(0). Then we discuss randomized competitive on-line algorithms that have been developed so far. We also mention lower bounds on the competitiveness that can be achieved by deterministic and randomized on-line algorithms. Finally we give a summary of average case analysis results for the list update problem.

In the context of self-organizing binary search trees we discuss restructuring heuristics such as Move-To-Root, Single-Rotation and Splaying. We then concentrate on the Splaying scheme and present important theorems known for splay trees. We also present a series of conjectures and report on progress on these conjectures. Finally we briefly mention how splay trees perform in practice.

In the third part of the talk we show that self-organizing binary search trees and in particular linear lists can be used to construct very effective data compression schemes.

Competitive Analysis of Distributed Algorithms
James Aspnes

Most applications of competitive analysis have involved on-line problems where a candidate *on-line* algorithm must compete on some input sequence against an optimal *off-line* algorithm that can in effect predict future inputs. Efforts to apply competitive analysis to fault-tolerant distributed algorithms require accounting for not only this *input nondeterminism* but also *system nondeterminism* that arises in distributed systems prone to asynchrony and failures. We survey recent efforts to adapt competitive analysis to distributed systems, and suggest how the results of these adaptations might in turn be useful in analyzing a wider variety of systems. These include tools for building competitive algorithms by composition, and for obtaining more meaningful competitive ratios by limiting the knowledge of the off-line algorithm.

On-line Load Balancing
YOSSI AZAR

We survey on-line load balancing on various models. The machine load balancing problem is defined as follows: There are n parallel machines and a number of independent tasks (jobs); the tasks arrive at arbitrary times, where each task has an associated *load vector* and duration. A task has to be assigned immediately to exactly one of the machines, thereby increasing the *load* on this machine by the amount specified by the corresponding coordinate of the load vector for the duration of the task. No admission control is allowed, i.e., all tasks must be assigned. The goal is to minimize the maximum load (or some other function of the load). We mainly consider non-preemptive load balancing, but in some cases we will allow preemption *i.e.,* reassignment of tasks. All the decisions are made by a centralized controller. The on-line load balancing problem naturally arises in many applications involving allocation of resources.

A Survey on Distributed Paging
YAIR BARTAL

In a distributed environment, data files are accessed for information retrieval by dispersed users and applications. The rapid development of the Internet, and Internet-related applications such as the World-Wide-Web, may be the most striking example.

Distributed data management problems are concerned with managing such distributed processing environments. Files can be replicated and allocated so as to reduce communication costs and to increase reliability while data consistency must be preserved.

We concentrate on three main problems. The file migration problem (Black and Sleator) allows only one copy of each file to be kept in the network. The file allocation problem (Bartal, Fiat and Rabani) deals with the more general case where files may be replicated and deleted in response to a sequence of read and write requests. The distributed paging problem (Bartal, Fiat and Rabani) further generalizes the model to deal with memory capacity limitations at the network processors. We give a survey of the various results on these problems.

The Wall Problem with Convex Obstacles
PIOTR BERMAN

We consider the problem of navigating through an unknown environment in which the obstacles are convex, and each contains a circle of diameter 1. The task is to reach a given straight line, in the distance n from our original position. Our randomized algorithm has competitive ratio $n^{0.75}$, and it uses tactile information only.

Joint work with Marek Karpinski.

On-Line Algorithms in Machine Learning

AVRIM BLUM

The areas of On-line Algorithms and Machine Learning are both concerned with problems of making decisions about a current situation based only on knowledge of the past. This survey talk discussed models, results, and open problems from Computational Learning Theory that seem particularly interesting from the point of view of on-line algorithms.

One Machine Learning problem with a flavor much like competitive analysis is that of learning a concept class C on-line in the presence of adversary noise, also known as "agnostic learning". Here, the goal is to achieve a prediction accuracy on an arbitrary sequence of examples that is nearly as good as that as that of the best concept from C. For the special case in which C is the class of single-variable concepts, also called the problem of "learning from expert advice", a performance ratio approaching 1 is achievable using algorithms of [Littlestone and Warmuth], [Vovk], and [Cesa-Bianchi et al.] Another problem with a flavor of competitive analysis is that of learning a target function that changes over time. For the case in which the target function is a disjunction, a simple modification of Littlestone's Winnow algorithm achieves a mistake bound of $O(\log n)$ times the number of variables that have been added to or removed from the target disjunction over time. This is a generalization of the result that Winnow makes at most $O(r \log n)$ mistakes when learning a fixed disjunction of r out of the n variables.

One open problem in this vein is the problem of learning Parity functions or Decision Lists with a mistake bound polynomial in the number of relevant variables r and only polylogarithmic in the total number of variables n. A potentially easier problem is that of learning either of these classes in the Infinite Attribute model [Blum, Hellerstein, Littlestone]. Another open problem is that of weak-learning disjunctions in the presence of 10% (or any constant fraction) adversarial noise.

Task Systems and the Server Problem

MAREK CHROBAK AND LAWRENCE L. LARMORE

This is an expository paper covering on-line algorithms for task systems and the server problem. We concentrate on results related to the so-called *work function algorithm* (WFA), especially its competitive analysis that uses the *pseudocost* approach.

After defining WFA and pseudocost, we show how to estimate the competitiveness of WFA in terms of pseudocost. Using this approach, we present the proof that WFA is $(2n-1)$-competitive for n-state metrical task systems and $(n-1)$-competitive for n-state metrical service systems (also called forcing task systems). We also show examples of metrical service systems in which WFA performs much worse than an optimally competitive algorithm.

We then turn our attention to the k-server problem. We present the proofs of the following results: that WFA is $(2k-1)$-competitive for k-servers, that it is 2-competitive for two servers, and k-competitive for metric spaces with at most $k+2$ points.

Variable sized bin packing and a dual version of bin packing

JANOS CSIRIK

In these two versions of the classical bin packing problem, we have a list $L = (a_1, a_2, ..., a_n)$ of elements with sizes $s(a_i)$ to pack. The aim of the packings is different: In *variable sized bin packing* we consider a finite collection of bins $B_1, B_2, ..., B_k$ (with sizes $s(B_i)$) and we want to pack the list into bins so as to minimize the total space (=total size of the used bins) used in the packing. Within the packing the total size of the elements in each bin can't be more than the size of the bin. In the *dual version of bin packing* (sometimes called bin covering) we have a bin capacity C for all bins and the aim is to pack the elements of L into a maximum number of bins so that the sum of the sizes in any used bin is at least C.

In this talk the best known on-line heuristics for these NP-hard problems are given. Lower bounds are also presented and some open problems are given.

Is it Rational to be competitive? — On the decision-theoretic foundations of the competitive ratio

RAN EL-YANIV

The competitive ratio, a performance measure for on-line algorithms, or alternatively, a decision making criterion for strict uncertainty conditions, has become a popular and accepted approach within theoretical computer science. In this talk we examine closely this criterion, both by characterizing it with respect to a set of axioms and in comparison to other known criteria for strict uncertainty.

Call admission algorithms that may delay calls

ANJA FELDMANN

New networking technologies such as ATM can carry a wide variety of traffic – voice, video, and data – over a single network. To provide the necessary bandwidth guarantees, the network should employ some mechanism to prevent links from becoming overly congested. This includes making intelligent decisions about whether to accept a call, and if so, when to schedule it, and how to route it. Such strategies are called call admission algorithms.

We present evidence that the burstiness of current traffic differs substantially from conventional traffic models. To cope with this burstiness and the high bandwidth requirements of multi-media applications we suggest call admission algorithms that may delay calls. Using competitive analysis and simulations to evaluate the algorithms we show that such call admission algorithms are an attractive alternative to those that may not delay calls.

On Multi-threaded Paging

ESTEBAN FEUERSTEIN

In this talk we introduce a generalization of Paging to the case where there are many threads of requests. This models situations in which the requests come from more than one independent source. Hence, apart from deciding *how* to serve a request, at each stage it is necessary to decide *which* request to serve among several possibilities.

Four different problems arise whether we consider fairness restrictions or not, with finite or infinite input sequences. We study all of them, proving lower and upper bounds for the competitiveness of on-line algorithms.

The main results presented in this paper may be summarized as follows. When no fairness restrictions are imposed it is possible to obtain good competitive ratios; on the other hand, for the fair case in general there exist no competitive algorithms.

Experimental Studies of Access Graph based Heuristics: Beating the LRU standard?

AMOS FIAT

We devise new paging heuristics motivated by the access graph model of paging [BIRS]. Unlike the access graph model [BIRS,IKP,FK] and the related Markov paging model [KPR], our heuristics are truly on-line in that we do not assume any prior knowledge of the program just about to be executed.

The Least Recently Used heuristic for paging is remarkably good, and is known experimentally to be superior to many of the suggested alternatives on real program traces [Y]. Experiments we've performed suggest that our heuristics beat LRU consistently, over a wide range of cache sizes and programs. The number of page faults can be as low as 1/2 less than the number of page faults for LRU and is on the average between 7–9 % better than LRU.

Our dynamic graph algorithm has a competitive ratio of $\Theta(k \log k)$, worse than that of LRU ($\Theta(k)$). In any case, the standard competitive ratio seems to be a poor measure of how well the algorithm really behaves. Hopefully, a better definition of an appropriate adversary generated sequence will give a better theoretical understanding of the problem.

We have built a program tracer that gives the page access sequence for real program executions of 200 – 1,500 thousand page access requests, and our simulations are based on these real program traces. While we have no real evidence to suggest that the programs we've traced are typical in any sense, we have made use of an experimental "protocol" designed to avoid experimenter bias.

We strongly emphasize that our results are only preliminary and that much further work needs to be done.

(Joint work with Ziv Rosen)

[**BIRS**] A. Borodin, S. Irani, P. Raghavan, and B. Schieber. Competitive Paging with Locality of Reference. In *Proc. 23rd Annual ACM Symposium on Theory of Computing*, pages 249–259, 1991.

[**FK**] A. Fiat and A.R. Karlin. Randomized and Multipointer Paging with Locality of Reference. In *Proc. 27th Annual ACM Symposium on Theory of Computing*, pages 626–634, 1995.

[**IKP**] S. Irani, A.R. Karlin and S. Phillips. Strongly competitive algorithms for paging with locality of reference. In *Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms*, 1992.

[**KPR**] A. R. Karlin, S. Phillips, and P. Raghavan. Markov paging. In *Proceedings of 33rd Annual Symposium on Foundations of Computer Science*, 1992.

[**Y**] N. Young, On-Line Caching as Cache Size Varies, Proceedings of the 2nd annual ACM-SIAM Symposium Discrete Algorithms, 1991, pp. 241–250.

Competitive Analysis of Paging: A Survey

SANDY IRANI

This talk surveyed results on the competitive analysis of paging. The speaker presented results showing tight bounds for the competitive ratio achievable by any on-line deterministic or randomized algorithm. The speaker then presented various refinements of the competitive ratio and the insights these models give to the paging problem.

On-Line Algorithms for Bin Packing and Scheduling: 1966 to the Present

DAVID S. JOHNSON

In the classical multiprocessor scheduling problem, one is given a number m of processors and a list of items (tasks) $a_1, a_2, \ldots, a_n$, each item having a size $s(a_i)$, and one is asked to assign the tasks to processors so as to minimize makespan, i.e., partition the items into m sets $P_1, \ldots, P_m$ so as to minimize the $\max_{1 \le j \le m} \sum_{1=1}^{m} s(a_i)$. In the classical bin packing problem, we assume that all sizes satisfy $s(a_i) \in (0,1]$

and ask for the minimum m such that the items can be scheduled on m processors with makespan 1 or less.

These two problems have had a central role in the history of on-line algorithm analysis. The competitive analysis of on-line algorithms appears to have started with R.L. Graham's 1966 paper on multiprocessor scheduling anomalies, and the first paper in which lower bounds were proved for the competitive ratio of any on-line algorithm was A.C. Yao's 1980 paper on new algorithms for bin packing. Some of the first work on average-case performance of on-line algorithms also occurred in the bin packing domain. This talk surveys the history of research on on-line algorithms for these two problems from the 1960's to the present. Both worst- and average-case results are covered.

A Survey of On-line Network Optimization Problems

Bala Kalyanasundaram and Kirk Pruhs

We survey results on on-line versions of the standard network optimization problems, including the minimum spanning tree problem, the minimum Steiner tree problem, the weighted and unweighted matching problems, and the traveling salesman problem. The goal in these problems is to maintain, with minimal changes, a low cost subgraph of some type in a dynamically changing network.

On the Performance in Practice of Competitive Algorithms

Anna R. Karlin

We present a brief survey of the results of empirical studies of competitive algorithms. We focus on four examples:

- applications of ski-rental including cache coherence, virtual circuit holding times, mobile computing and dynamic compilation,
- IP paging,
- routing and admission control, and
- dynamic data structures such as splay trees.

These performance studies lead to the following conclusions: First, algorithms optimized to work against the worst-case adversary tend not to work well in practice. Second, the standard competitive algorithms give insight leading to the "correct" algorithm. Finally, nearly all successful empirical studies use algorithms that adapt in some way to the input distributions.

On-Line Graph Coloring

Henry A. Kierstead

We present a survey of three types of results concerning on-line graph coloring. There first type deals with the problem of on-line coloring k-chromatic graphs on n vertices, for fixed k and large n. The second type concerns fixed classes of graphs whose on-line chromatic number can be bounded in terms of their clique number. Examples of such classes include interval graphs and the class of graphs that do not induce a particular radius two tree. The last type deals with classes of graphs for which First-Fit performs reasonably well in comparison to the best on-line algorithms. Examples of such classes include interval graphs, the class of graphs that do not induce the path on five vertices, and d-degenerate graphs.

A new on-line graph coloring algorithm

Henry A. Kierstead

We prove the following theorem: For every positive integer k, there exist positive constants C and ϵ and an on-line algorithm A such that for all k-colorable graphs G on n vertices and all input sequences $v_1 < \ldots < v_n$, A uses at most $C * n^{1-\epsilon}$ colors to properly color G.

A Competitive Strategy for Learning an Unknown Simple Polygon

Rolf Klein

Suppose that a point-shaped mobile robot, equipped with an on-board 360° vision system, has to learn a simple polygon. To this end, the robot starts from a fixed point x_0 on its boundary, moves around inside the polygon, and returns to x_0. On its way, each point of the polygon must be visible at least once. The length of the robot's path is compared against the length of the optimum watchman tour that runs inside the polygon, contains x_0, and sees every point.

In their FOCS'91 paper, Deng, Kameda, and Papadimitriou were the first to claim that for this problem a competitive strategy exists. For the special case of rectilinear polygons, they proved in 1993 that even a simple greedy strategy achieves a factor of $\sqrt{2}$.

We present the first competitive strategy for the more difficult case of general polygons. Our strategy explores the polygon room by room, in depth-first order. A room R_x is a subpolygon associated with an entry vertex, x. With respect to the shortest paths from x at most one change between left and right vertices can occur within R_x. If x itself is a right vertex then first all right descendants of x are explored, in clockwise order, on a tour that starts from and returns to x. On a second tour, the left descendants of x and of its right offspring are explored in counterclockwise order. Afterwards, the robot has completely seen R_x and knows about the adjacent rooms.

We can prove that the path created by our strategy does not exceed in length 133 times the length of the optimum watchman tour. However, the size of this bound seems to be caused by the analysis, rather than by the actual performance of our strategy; we conjecture its true competitive ratio migth be less than 20.

Joint work with Frank Hoffmann (FU Berlin), Christian Icking (FernUni Hagen), and Klaus Kriegel (FU Berlin).

Limited randomization in on-line computation

ELIAS KOUTSOUPIAS

So far, there has been little progress in analyzing randomized algorithms against oblivious adversaries. We propose a way to overcome the difficulties by focusing our attention on randomized on-line algorithms with limited randomization. These algorithms choose initially an algorithm among a fixed finite set of deterministic algorithms, and then simply simulate it. The competitive analysis of such a randomized algorithm for a problem A corresponds naturally to the analysis of a deterministic algorithm for another problem A'. A typical example is the 1-evader problem, the equivalent of the k-server problem for $k+1$ points. For example, the competitive ratio for the 1-evader problem with one random bit is equal to the competitive ratio of the unrestricted 2-evader problem. We also discuss certain bounds between randomized and deterministic competitive ratios.

On-Line Routing in All-Optical Networks

STEFANO LEONARDI

This work deals with on-line routing in WDM (wavelength division multiplexing) optical networks. A sequence of requests arrives over time, each is a pair of nodes to be connected by a path. The problem is to assign a wavelength and a path to each pair, so that no two paths sharing a link are assigned the same wavelength. The goal is to minimize the number of wavelengths used to establish all connections. This problem has been introduced by Raghavan and Upfal in 1994 that started the study of the off-line version of this problem. For a line topology, the problem is just an interval graph coloring problem. On-line algorithms for this problem have been analyzed by Kierstead and Trotter since 1981.

We consider trees, trees of rings, and meshes topologies, previously studied in the off-line case. We give on-line algorithms with competitive ratio $O(\log n)$ for all these topologies. We give a matching $\Omega(\log n)$ lower bound for meshes. We also prove that any algorithm for trees cannot have competitive ratio better than $\Omega(\frac{\log n}{\log\log n})$.

We also consider the problem where every edge is associated with parallel links. While in WDM technology, a fiber link requires different wavelengths for every transmission, SDM (space division multiplexing) technology allows parallel links for a single wavelength, at an additional cost. Thus, it may be beneficial in terms of network economics to combine between the two technologies (this is indeed done in practice). For arbitrary networks with $\Omega(\log n)$ parallel links we give an on-line algorithm with competitive ratio $O(\log n)$.

Joint work with Yair Bartal.

Computational Aspects of Organization Theory

Christos Papadimitriou

In this talk I review the use of concepts and techniques from the fields of on-line algorithms and combinatorial optimization in order to gain insight into the problem of optimally organizing the flow of information and authority in a group of decision makers in order to achieve high-quality decisions. We consider two kinds of obstacles: *incomplete information* and, perhaps more subtle, *different objectives*, that is, situations in which various decision-makers optimize different, and sometimes conflicting, functions.

On-line Routing and Admission Control in High-Speed Networks

Serge Plotkin

Emerging high speed Broadband Integrated Services Digital Networks (B-ISDN) will carry traffic for heterogeneous services such as video-on-demand and video teleconferencing. These services require resource reservation along the path on which the traffic will travel. Current proposals call for reserving the resources in terms of Virtual Circuits.

The large bandwidth-delay product of broadband networks prevents rerouting of existing circuits. Thus, routing decisions (i.e. choices of paths) have to be done carefully, to make sure that the bandwidth is used in the most efficient way. Since network resources are limited, there is a need for admission control. Routing and admission control decisions have to be made on-line, without any knowledge of the future requests.

The talk surveys several recently developed routing and admission control algorithms and present results of a simulation study of these algorithms on several topologies. These algorithms draw on techniques developed in the context of approximation algorithms for multicommodity flow and on-line algorithms, together with stochastic analysis developed in the context of “trunk-reservation” algorithms. Simulation studies show that our algorithms significantly outperform the more traditional greedy approaches on a commercial network topology and several of its variants.

On Call-Admission in Optical Networks

Adi Rosén

In this short talk we describe known results on competitive call admission in communication networks, comparing high-speed networks to optical networks, and describe an open problem related to those.

While in high-speed networks the problem is that of maximizing the number of calls accepted under the link capacity constraints, in optical networks the constraint is that edge-disjoint paths must be maintained for each wavelength. For the case of high speed networks, the problem becomes much easier if the capacity of each link is at least $\Omega(\log n)$, where n is the size of the network [AAP, FOCS 93], resulting in competitive ratios smaller than the lower bound for the on-line edge-disjoint paths problem. Call admission in optical networks can be reduced to the problem of edge-disjoint paths maintaining almost the same competitive ratio [AAFLR, ESA 96]. However, it is an open problem whether for the case of optical networks a certain number of parallel wavelengths allows for competitive ratios smaller than the lower bound for the on-line edge disjoint path problem.

Short Paths on Expander Graphs

RONITT RUBINFELD

Graph expansion has proved to be a powerful general tool for analyzing the behavior of routing algorithms and the inter–connection networks on which they run. We develop a number of new routing algorithms and structural results for bounded–degree expander graphs. Our results are unified by the fact that they are all based upon, and extend, a body of work asserting that expanders are rich in short, disjoint paths.

In particular, our work has consequences for the disjoint paths problem, multicommodity flow, and graph minor containment. We show:

(i) A greedy algorithm for approximating the maximum disjoint paths problem achieves a polylogarithmic approximation ratio in bounded–degree expanders. Our approximation ratio is an improvement over the current best bounds for the problem in expanders. This is interesting in particular because our algorithm, unlike previous polylogarithmic approximations, is both *deterministic* and *on–line*.

(ii) For a multicommodity flow problem with arbitrary demands on a bounded–degree expander, there is a $(1+\epsilon)$–optimal solution that uses exclusively flow paths of polylogarithmic length. It follows that the recent multicommodity flow algorithm of Awerbuch and Leighton runs in nearly *linear* time per commodity in a bounded–degree expander. Our analysis is based on a combinatorial problem that arises from the linear programming dual of the multicommodity flow. We show that for any choice of edge weights on any expander G, one can increase some of the weights very slightly so that the resulting shortest–path metric is *smooth* — the minimum–weight path between any pair of nodes uses only a polylogarithmic number of edges.

(iii) Every bounded–degree expander on n nodes contains every graph with $O(n/\log^{O(1)} n)$ nodes and edges as a minor.

Joint work with Jon Kleinberg.

Lower Bounds in On-line Geometric Searching

SVEN SCHUIERER

We present a technique to prove lower bounds for on-line geometric searching problems. It is assumed that a goal which has to be found by a searcher is hidden somewhere in a known environment. The search cost is proportional to the distance traveled by the searcher. We are interested in lower bounds on the *competitive ratio*, that is the ratio of the distance traveled by the searcher to the shortest possible distance to reach the goal.

The technique we present is based on an observation about the geometry of high-dimensional convex cones. It can be applied whenever the strategies for the search problem can be represented as sequences of positive numbers and the competitive ratio can be expressed as the maximum of a sequence of functionals which satisfy certain requirements. As an application we show that there is no strategy for searching on two rays whose average competitive ratio taken over the competitive ratios for each ray is better that the overall competitive ratio.

On-Line Scheduling — A Survey

JIŘÍ SGALL

In the classical scheduling problems we have some sequence of jobs that have to be processed on the machines available to us. In the most basic problem, each job is characterized by its running time and has to be scheduled for that time on one of the machines. In other variants there may be additional restrictions or relaxations as to which schedules are allowed. We want to schedule the jobs as efficiently as possible, which most often means that the total length of the schedule (the makespan) should be as small as possible, but other performance measures are also considered.

The notion of an on-line algorithm is intended to formalize the realistic scenario, where the algorithm does not have the access to the whole input instance, unlike the off-line algorithms. Instead, it learns the input piece by piece, and has to react to the new requests with only a partial knowledge of the inputs. Such scheduling algorithms are the topic of this survey.

On-line Scheduling Algorithms to Minimize the Average Completion Time

DAVID B. SHMOYS

We consider the following on-line scheduling model: jobs arrive over time, and for each point in time, the schedule up to that point must be determined without knowledge of future arrivals. Each job j has a specified weight w_j and a specified processing time p_j that are known at its arrival time; the aim is to schedule the jobs so as to minimize $\sum w_j C_j$, where C_j denotes the time at which job j completes processing.

We give a general technique that yields strong performance guarantees in a wide variety of scheduling environments. This approach shows that a certain off-line subroutine is sufficient for obtaining good on-line performance. Furthermore, there is a simple randomized variant that yields stronger bounds (in expectation). For example, even for the simplest case in which there is a single machine, this framework yields an algorithm with the strongest known performance guarantee, even in the off-line setting. We shall discuss several other applications of this technique, including the scheduling of both malleable and non-malleable jobs on parallel machines.

This is joint work with Soumen Chakrabarti (U.C./Berkeley), Leslie Hall (Johns Hopkins University), Cynthia Phillips (Sandia National Labs), Andreas Schulz (TU-Berlin), Cliff Stein (Dartmouth), and Joel Wein (Polytechnic University).

My Adventures in On-Line Algorithms

Daniel Sleator

In this talk I trace the development of my thinking on what is now called competitive analysis of algorithms.

The path splicing problem arises when attempting to devise efficient data structures for the "dynamic tree" problem. (Which itself arises from the maximum flow problem.) An elegant solution to the path splicing problem using a potential function illustrated the power of this approach. The potential fnction method was further developed by Bob Tarjan and me in self-adjusting data structures, like splay trees and skew heaps. Our analysis of the move to front heuristic introduced the notion of comparing the performance of an on-line algorithm to the off-line optimum in a data structure.

This style of thinking was used by Anna Karlin, Mark Manasse, Larry Rudolf, and me to analyze the snoopy cashing problem. Manasse coined the term "competitive".

Mark Manasse, Lyle McGeoch and I proposed and gave a preliminary analysis of "server problems".

This "competitive" approach to analyzing on-line problems has driven the development of new algorithms for a variety of problems, and it is a fertile source of mathematical problems.

On-line Traveling Salesman Problems

Leen Stougie

We consider a class of on-line variations of TSP in a metric space: while the salesman is traveling, new sites to visit may be communicated to him. His goal is to visit all the sites and eventually return to the departure point, minimizing the completion time.

We present an optimal algorithm that is 2-competitive on any metric space for the version of the problem in which it is required to return back to the departure point after all presented points have been visited. In case the metric space is the real line we present a 1.75-competitive algorithm that compares with a ≈ 1.64 lower bound.

We also consider a version of the problem in which the constraint of returning to the departure point is dropped. For this problem we derive a lower bound on the competitive ratio of 2. This lower bound is achieved via a problem on the real line. Besides, a 2.5-competitive algorithm for any metric space and a 7/3-competitive algorithm for the real line are provided.

Interesting problems that remain to be solved in the context of this problem are to close the gaps between lower and upper bounds on the competitive ratio's that still exist, and to see if randomization might be of any help in the model where the server is to return to the starting point (for the other model we have proved that also any randomized algorithm can be no better than 2-competitive).

Another interesting version of the problem is to minimize the sum of the waiting times of the points (customers) to be served, i.e., the difference between the time they are presented and the time they are visited.

This is joint work with Giorgio Ausiello, Esteban Feuerstein, Stefano Leonardo, and Maurizio Talamo.

Disjoint Paths in Densely Embedded Graphs

EVA TARDOS

We consider the following maximum disjoint paths problem. We are given a large network, and pairs of nodes that wish to communicate over paths through the network — the goal is to simultaneously connect as many of these pairs as possible in such a way that no two communication paths share an edge in the network. This classical problem has been brought into focus recently in papers discussing applications to routing in high-speed networks, where the current lack of understanding of the maximum disjoint paths problem is an obstacle to the design of practical heuristics.

We consider the class of *densely embedded, nearly-Eulerian graphs*, which includes the two-dimensional mesh and many other planar and locally planar interconnection networks. We obtain a constant-factor approximation algorithm for the maximum disjoint paths problem for this class of graphs; this improves on an $O(\log n)$-approximation for the special case of the two-dimensional mesh due to Aumann–Rabani and the authors. For networks that are not explicitly required to be "high-capacity," this is the first constant-factor approximation for the maximum disjoint paths problem in any class of graphs other than trees.

We also consider the maximum disjoint paths problem in the on-line setting, relevant to applications in which connection requests arrive over time and must be processed immediately. We obtain an asymptotically optimal $O(\log n)$-competitive on-line algorithm for the same class of graphs; this improves on an $O(\log n \log\log n)$-competitive algorithm for the special case of the mesh due to Awerbuch, Gawlick, Leighton, and Rabani.

The talk presents joint work with Jon Kleinberg.

Randomized Weighted Caching and Free Time
Andrew Tomkins

We consider *weighted cache* spaces, in which each point i has an associated weight w_i, and the distance between points is given by $d_{i,j} = w_j$. We give an $O(\log^2 k)$-competitive randomized algorithm for $k+1$-point weighted cache spaces, and an $\Omega(\log k)$ lower bound on the competitive ratio of any algorithm on any such space. These results also apply to any metrical task system on spaces corresponding to weighted star graphs.

We also consider an extension of the standard on-line model to settings in which an on-line algorithm has free time between successive requests in an input sequence. During this free time, the algorithm may perform operations without charge before receiving the next request. For instance, in planning the motion of fire trucks, there may be time in between fires that one could use to reposition the trucks in anticipation of the next fire. We prove both upper and lower bounds on the power of deterministic and randomized algorithms in this model.

We also consider variants of the free-time model in which both free time and hints about upcoming requests are available, and variants in which the available free time is bounded.

This is joint work with Avrim Blum and Merrick L. Furst.

Semi on-line algorithms for the partition problem
Zsolt Tuza

Let $S = \{s_1, \ldots, s_n\}$ be a multiset of natural numbers, $[n] := \{1, 2, \ldots, n\}$, and denote $w(Y) := \sum_{i \in Y} s_i$ for all $Y \subseteq [n]$. The optimization version of the partition problem is to determine

$$w_{opt} := \min_{Y_1 \cup Y_2 = [n]} \max\left(w(Y_1), w(Y_2)\right).$$

The problem of deciding whether $w_{opt} = \frac{1}{2} w([n])$ is well know to be NP-complete (Karp, 1972).

Assume that an algorithm A generates the partition with classes Y_A and $Y'_A = [n] \setminus Y_A$. Denoting $w_A := \max\left(w(Y_A), w(Y'_A)\right)$, the ratio w_A/w_{opt} is termed the competitive ratio of A. It has been proved by Graham [*Bell Syst. Techn. J.*, 1966] that the smallest possible competitive ratio of an *on-line* partition algorithm is 3/2.

We consider the following three types of semi on-line algorithms:

- the total sum $w([n])$ of the items is known;
- two parallel processors are available, each producing a partition on-line, and the best of the two is chosen at the end;
- a buffer of fixed size $k \geq 1$ is available where k items may be stored for unlimited time.

We prove that, in each of these cases, the best possible algorithms have competitive ratio 4/3 (independently of the value of k in the third variant). This tight bound also happens to be the same as the one for randomized on-line algorithms, by a theorem of Bartal et al. [*J. Comput. Syst. Sci.*, 1995].

On-line scheduling on parallel machines

Arjen P.A. Vestjens

We consider parallel machine scheduling problems where jobs arrive over time. A set of independent jobs has to be scheduled on m machines, where the number of jobs is unknown in advance. Each job becomes available at its release date, which is not known in advance, and its processing requirement becomes known at its arrival. We deal with the problem of minimizing the makespan, which is defined as the time by which all jobs have been finished.

If we consider the machines to be identical and allow preemption, then the problem can easily be solved by a generalization of McNaughton's wrap-around rule. This result is due to Hong and Leung. We consider two models that are closely related to this model.

In the first model, we consider uniform machines, i.e., the machines have different speeds for processing the jobs, and allow preemption. We show that if only a finite number of preemptions is allowed, there exists an on-line algorithm that solves the problem if and only if $s_{i-1}/s_i \leq s_i/s_{i+1}$ for all $i = 2, ..., m-1$, where s_i denotes the i-th largest machine speed. We also show that if this condition is satisfied, then $O(mn)$ preemptions are necessary, and we provide an example to show that this bound is tight.

In the second model, we consider identical machines again, but forbid preemption. We analyze the following on-line LPT algorithm: At any time a machine becomes available for processing, schedule an available job with the largest processing requirement. We prove that this algorithm has a performance guarantee of 3/2. Furthermore, we show that any on-line algorithm will have a performance bound of at least 1.3473. This bound is improved to 1.3820 for m=2.

Providing Good Average Service to Jobs That Arrive Over Time

JOEL WEIN

One of the most natural questions in on-line scheduling is to provide good average service to a stream of jobs that arrive, in an unannounced fashion, over time. This is typically modeled as the problem of scheduling jobs with *release dates* so as to minimize *average completion time* or *average flow time.*

Recently there has been significant progress on developing small-constant factor approximation algorithms to minimize average completion time on a single or m parallel machines. There has also been progress in understanding the approximability of the average flow time criterion.

We review a number of recent results in this area and present some very recent results on on-line algorithms for average flow time in a relaxed measure of competitiveness, recently introduced by Kalyanasundaram and Pruhs, in which the on-line algorithm is allowed more resources than the off-line algorithm.

The results discussed are largely joint work with various subsets of Leslie Hall, Rajeev Motwani, Cindy Phillips, Andreas Schulz, David Shmoys, Cliff Stein, and Eric Torng.

Open Problems

An open problem on Uniform Service Systems with k-servers
Esteban Feuerstein

In [CL], Chrobak and Larmore proposed a family of on-line problems, namely Metrical Service Systems (MSS). In an instance of MSS_w *one* server situated on a metric space must serve a sequence of requests, where each request consists of a set of at most w nodes of the metric space, and can be served by moving the server to *any* of the nodes of the set. The goal is to minimize the total distance traveled by the server. An important particular case of MSS is when the metric space is uniform, i.e. when all the distances are equal (uniform-MSS). Both MSS and uniform-MSS are particular cases of Metrical Task Systems, but not of the k-server problem, as each request specifies different alternative nodes to cover.

In [F95,F96] we present the generalization of uniform-MSS_w to the case in which $k \geq 1$ servers are used. We call this problem (k, w)-Uniform Service Systems (abbreviated as $\mathrm{USS}_{(k,w)}$).

It is a well known fact that the k-server problem on uniform metric spaces is isomorphic to the paging problem with a cache of size k. In the same way, $\mathrm{USS}_{(k,w)}$ can be seen as the following "Paging" problem: Given a universe U, an on-line algorithm with a cache of size k must deal with a finite sequence of requests, each of which consists in a subset $r \subset U$ of size at most w. Each request is served by having in the cache at least one element of r.

In [F95,F96] it is shown that no on-line algorithm for $\mathrm{USS}_{(k,w)}$ can achieve a competitive ratio better than $\binom{k+w}{w} - 1$, and we present an $O(k \min(k^w, w^k))$-competitive algorithm. For any fixed value of k (and arbitrary w) this is at most a constant factor away from the lower bound . However, for k tending to infinity the upper bound is a constant times k away from optimality. We conjecture that the same algorithm achieves a competitive ratio of $O(\min(k^w, w^k)$ (and therefore is only a constant factor away from optimal also when k tends to infinity), but we have not proved it in the general case. Our algorithm solves at each step an instance of *Hitting-set*, a well known NP-complete problem.

[CL] M. Chrobak and L. Larmore. The server problem and on-line games. In *On-line Algorithms*, volume 7 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 11–64. AMS-ACM, 1992.

[F95] E. Feuerstein. On-line Paging of Structured Data and Multi-Threaded Paging. Ph.D. Thesis, Università di Roma "La Sapienza", 1995.

[F96] E. Feuerstein. Uniform Service Systems with ¬-servers. Unpublished manuscript.

A generalization of approximation factors and the competitive ratio

AMOS FIAT

One of the many problems with the competitive ratio is that one gets very bad (very high) competitive ratios for important problems. One could make a similar observation about off-line approximations of problems such as max clique and independent set.

Consider the problem of edge-disjoint paths, in this problem we are given an on-line set of pairs of vertices, for every pair we have to decide if to accept or reject it, and if we accept a pair we need to display an edge-disjoint path between the two vertices. Non-trivial polylogarithmic upper bounds are possible in certain cases [AGLR,KT], but there is an n^ϵ lower bound for all randomized algorithms against an oblivious adversary in general graphs [BFL]. A problem with the lower bound of [BFL] is that it holds only when the set of paths accepted by the adversary is relatively small (an n^δ fraction of the total number of pairs of vertices presented). One could argue that this is a non-interesting lower bound because it only applies when the total adversary benefit is very small (although in this case the on-line algorithm does even worse).

A phone company that were to accept only a small fraction of all call requests would not stay in business for long. Thus, it is “unfair” to consider a lower bound on the competitive ratio when the adversary benefit is small. Consider the independent set approximation problem, this problem cannot be well approximated [FGLSS]. But, if we know that 98% of the vertices are independent it is trivial to obtain an independent set with 96% of the vertices.

I.e., what we suggest is to use a new parameter for online approximation problems: how good (or how large) is the optimal solution when compared to the instance size. Ideally, one could perhaps come up with approximation algorithms that depend on this parameter, at worst they would be identical to the general lower bounds, but they could be much better when this parameter is large (or small).

It would be of interest to display problems for which the competitive ratio is poly logarithmic if the adversary benefit is large, but increases to some n^ϵ when the adversary benefit drops. A trivial example where one can show dependence on this parameter is that of on-line matching. Perhaps most interesting would be to show that this parameter is of significance in the context of on-line edge disjoint paths, the ultimate routing problem.

[AGLR] B. Awerbuch, R. Gawlick, F.T. Leighton and Y. Rabani. On-line Admission Control and Circuit Routing for High Performance Computing and Communication. In *Proc. of the 35th Annual Symposium on Foundations of Computer Science*, 1994.

[BFL] Y. Bartal, A. Fiat, and S. Leonardi. Lower Bounds for On-line Graph Problems with Application to On-line Circuit and Optical Routing. Proceedings of the 28th annual ACM symposium on the Theory of Computing, May 1996, pages 531–540.

[FGLSS] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Approximating

Clique is almost *NP*-complete. Proc. 32nd IEEE Symp. on the Foundations of Computer Science, pages 2–12, 1991.

[KT] J. Kleinberg and E. Tardos. Disjoint Paths in Densely Embedded Graphs. *Proc of the 36-th IEEE Annual Symposium on Foundations of Computer Science*, 1995.

A server problem due to Rajeev Motwani

AMOS FIAT

Consider the set of points on a line: $0, 1, 2, \ldots, n-1$. An on-line event sequence σ consists of two types of events:

- service(p), where $p \in \{0, 1, \ldots, n-1\}$.
- charge-per-server.

An algorithm for this problem (on-line or off-line) has a dynamically changing set of servers located at some subset of $\{0, 1, \ldots, n-1\}$. The number of servers may vary over time, but there are never less than one nor more than n servers at any given point of time. Following a service(p) request, some server must be located at point p to service the request. Servers do not move, but new servers can be born adjacent to old servers: Given a server at point q, a new server can be located at one of the two adjacent points, $q \pm 1$, at a cost of 1.

Consider the configuration in which there is a server a point $q < p$ but no server in the range $q+1, \ldots, p$. One possible way to service the request service(p) is to generate $p-q$ new servers, one at each point $q < i \leq p$, for a total cost of $p-q$. When a charge-per-server event arrives, the algorithm first has the choice of deleting any number of servers (but must leave at least one server), at a cost of zero. Following this deletion, the algorithm then pays 1 for every remaining server.

A natural algorithm for the problem is as follows: For a service(p) — replicate servers from the closest current server to p. For a charge-per-server event — delete every 2nd server from left to right, except for the last server. An obvious $\sqrt{n}$ competitive on-line algorithm is to leave servers at points 0, $\sqrt{n}, 2\sqrt{n}, \ldots, n-\sqrt{n}$ following every charge-per-server event, and to replicate servers from the closest server at service requests. What is the competitive ratio for the "natural algorithm" above? What is the competitive ratio for the problem?

Call control with edges of capacity two

STEFANO LEONARDI

The *Call Control* problem with edge-capacity 2 is the following: An undirected network $G = (V, E)$ is known to the algorithm, at every step a request to establish a virtual circuit between two nodes of the network is presented to the on-line algorithm. The algorithm may accept the request and then assign a path connecting the two nodes, or reject the request. The constraint is that no more than two paths contain a given edge. The goal is to maximize the number of accepted requests.

If the edges have capacity 1 (the on-line edge-disjoint paths problem), then there is an $\Omega(n^\epsilon)$ lower bound on the competitive ratio of randomized algorithms [BFL], where n is the number of nodes in the network. This lower bound is obtained on a network that admits an $O(\log n)$-competitive randomized algorithm if edges have capacity 2 [KT]. The open question is: Does a polylogarithmic competitive randomized algorithm for Call Control with capacity 2 on any network exist, or there is a network with an $\Omega(n^\epsilon)$ lower bound on the competitive ratio of randomized algorithms? The problem admits a deterministic $O(\log n)$ competitive algorithm if edges have logarithmic capacity [AAP]

[AAP] B. Awerbuch, Y. Azar and S. Plotkin. Throughput Competitive On-line Routing. *Proc. of the 34th Annual Symposium on Foundations of Computer Science*, November 1993.

[BFL] Y. Bartal, A. Fiat, and S. Leonardi. Lower Bounds for On-line Graph Problems with Application to On-line Circuit and Optical Routing. Proceedings of the 28th annual ACM symposium on the Theory of Computing, May 1996, pages 531–540.

[KT] J. Kleinberg and E. Tardos. Disjoint Paths in Densely Embedded Graphs. *Proc of the 36-th IEEE Annual Symposium on Foundations of Computer Science*, 1995.

Scheduling of Loops in Multiprocessor Systems

JOEL WEIN

Consider the scheduling of N independent iterations of a loop on a parallel multiprocessor with p processors. In practice, iterations are distributed from a global queue. If N/p iterations are given to each processor, we have low overhead but poor load balance; if we give out one iteration at a time we have good load balance but huge overhead. A compromise is to have the global queue distribute the iterations in chunks of some fixed size, but an even better solution is to decrease the size of the chunk over time. This method has been established to be excellent in practice, and has some analysis when a distribution is assumed about the iteration size. **Is there some competitive analysis that differentiates between these strategies?** We certainly need to expand the models of on-line scheduling to include the overhead of going to the queue, and perhaps assume some partial information prior knowledge. (This problem arose in discussions with Bill Aiello, Sandeep Bhatt, and Susan Flynn Hummel).

A good reference is: S. Flynn Hummel, E. Schonberg, and L. E. Flynn. Factoring: A Practical and Robust Method for Scheduling Parallel Loops. *Comm. of the ACM* **35(8)**, 90–101, 1992.