SIGACT News Online Algorithms Column 30: 2016 in review

Rob van Stee
University of Leicester
United Kingdom

In this column, I will discuss some recent papers in online algorithms that appeared in 2016. Quite a number of papers on online algorithms appeared during the year, and I have had to make a selection; I am sure to have overlooked some papers. If I have unaccountably missed your favorite paper and you would like to write about it or about any other topic in online algorithms, please don't hesitate to contact me!

1 Online packing and covering with convex objectives

In the most high-profile result of the year and (I assume) the one with the most authors (eleven), Azar et al. [6] give a general framework for solving packing and covering problems with convex objectives (instead of linear). This FOCS 2016 paper combines the results of three research groups and allows many existing results to be unified, simplified and/or improved.

To begin with, this improves and generalizes previous results for problem classes such as mixed packing covering LPs considered earlier [5]. The previous result for minimizing the maximum constraint was a bound of $O(\log K \log(d\kappa\gamma))$, where K is the number of objective functions, d is the row-sparsity of the covering constraint matrix A, κ is the ratio between the maximum and the minimum entries in the packing constraints, and γ is the ratio between the maximum and the minimum entries in the covering constraints. This is now improved to an $O(\log K \log d)$-competitive algorithm.

The authors show how to round the obtained fractional solutions online for specific optimization problems, obtaining improved results for non-linear optimization problems considered recently in machine scheduling, network routing, and combinatorial auctions. Examples of the problems it applies to are social welfare maximiation with production costs, unrelated machine scheduling with startup costs, capacity constrained facility location and online set cover with set requests.

2 Steiner trees and Steiner forests

There is a long history of research into the problem of satisfying connectivity demands on a graph while respecting given constraints, going back to the early seventies. There are many variations, but probably the best known version is the **Steiner tree problem**, in which a set of vertices in a graph need to be connected by a tree (a subset of the edges of the graph) of minimum weight. In the online version, the graph is still given in advance, but the vertices which need to be connected are given one by one.

On arbitrary graphs, a greedy (nearest neighbor) algorithm is $O(\log n)$-competitive, where n is the number of nodes. It is not possible to do better even on series-parallel graphs. In WAOA 2016, Akira Matsubayashi [25] gives an 8-competitive deterministic algorithm for outerplanar graphs. Previously, constant competitive algorithms were known only for trees and rings. The algorithm works by not using the shortest path to connect a new node, but rather by finding a path that is a constant times longer than the shortest path.

Degree-bounded Steiner forest problem A variation of the Steiner tree problem is to ask for a tree of *minimum degree* which connects the vertices. In the more general degree-bounded Steiner forest problem (DB-SF), the demands consist of vertex pairs, and the goal is to output a subgraph (a forest) in which for every demand there is a path connecting the pair.

In SODA 2016, Dehghani et al. [17] initiate the study of the online version of this problem, where connectivity demands appear over time and must be immediately satisfied. They show that a greedy algorithm, which adds paths in such a way as to minimize the maximum load on vertices of the new path, is $O(\log n)$-competitive. The load of a vertex is defined as the degree of the vertex in the currently selected solution divided by the degree bound of the vertex, which is given in advance. Again, this algorithm is shown to be best possible, by proving that even randomized algorithms for the Steiner tree problem cannot do better.

Edge-weighted degree-bounded Steiner forest The same paper also considers the edge-weighted version (EW-DB-SF). The authors show that a (randomized) online algorithm cannot be $o(n)$-competitive with respect to the maximum degree and $o(n)$-competitive with respect to the weight at the same time. However, they also point out that this result does not rule out the existence of a competitive algorithm if there is a bound on the edge weights. Indeed, in a followup paper in ICALP 2016, Dehghani et al. [16] (a slightly expanded set of authors) show that it is possible to be $O(\log^3 n)$-competitive with respect to the weight and $O(\log^3 n \log n\rho)$-competitive with respect to the maximum degree. Here ρ is defined as the ratio between the maximum weight and the minimum nonzero weight of any edge. It is possible to improve the competitiveness with respect to the degree slightly, at the cost of a slightly worse ratio with respect to the weight. Of course, given the above lower bound, it is unavoidable to have a dependence on ρ.

As mentioned above, Azar et al. [5], building on earlier work [3, 10], gave a framework for solving fractional mixed packing and covering LPs, and this could be used to give a fractional solution with polylogarithmic competitive ratio for EW-DB-SF. Dehghani et al. [16] note that doing the rounding in an online manner seems very hard. The problem is that the underlying fractional solution may change drastically in between two rounding steps.

Dehghani et al. circumvent this problem by formulating EW-DB-SF as an IP which has exponential size but in which each variable occurs only once in the covering constraints. They then design a new integral algorithm for solving this restricted family of IPs, which may of course have applications elsewhere. Their main result for this family is the following.

Theorem 1 *[16] Given an instance of the online mixed packing/covering integer program, there exists a deterministic integral algorithm with competitive ratio $O(k \log m)$, where m is the number of packing constraints and k is the covering frequency of the IP.*

Here the covering frequency is defined as the maximum over all variables of the number of covering constraints that contains a given variable. They furthermore show that even randomized, fractional algorithms cannot improve on this result.

3 Online scheduling

Several papers on online scheduling appeared last year. To begin with the last one, Böhm et al. [8] considered the **online packet scheduling problem**, where packets of unit size arrive at a router over time and need to be transmitted over a network link. Each packet has a weight and a deadline, and the competitive ratio for this problem is somewhere between $\phi \approx 1.618$ [4, 13, 22] and 1.828 [19]. There has been no improvement in these bounds for ten years now.

A special case is that of s-bounded instances. Here each packet has a relative deadline of s when it arrives, so it can only be scheduled in one of the first s slots after its arrival. The lower bound of ϕ applies to this case as well, and a ϕ-competitive algorithm for 3-bounded instances was given by Michael Goldwasser in this very column in 2010 [21]. Böhm et al. [8] improve upon this result by giving a ϕ-competitive algorithm for 4-bounded instances.

The algorithm is a modification of the well-known Earliest Deadline First (EDF) algorithm, which sometimes selects a packet that is lighter than EDF would select. Their algorithm uses memory, by marking one pending packet under certain circumstances, and the authors leave it as an open question whether this result can be achieved without memory.

It should be noted that even for s-bounded instances, no better upper bound than 1.828 is known (for general s). This remains a fascinating open problem!

Machine minimization In SODA 2016, Chen, Megow and Schewior [11] gave an $O(\log m)$-competitive algorithm for the online machine minimization problem. Here the goal is to find a feasible preemptive schedule for jobs with hard deadlines which arrive online on a minimum number of machines. This is the first improvement on this problem since 1997, when Phillips et al. [26] gave an $O(\log \Delta)$-competitive algorithm, where Δ is the ratio between the maximum and the minimum possible job size. The advantage of the new algorithm is that the result does not depend on Δ and that a competitive ratio of $O(1)$ is achieved for laminar and agreeable instances.

The new result is achieved by deriving a new lower bound on the optimal solution value, which relates the laxity and the number of jobs with intersecting time windows.

The same authors consider the power of migration in a followup paper [12] which appeared in SPAA 2016. Here migration refers to continuing a job on a different machine after preempting it. They show that an algorithm which is not allowed to use migration cannot be competitive as a function of the number of machines, and must be $\Omega(n)$-competitive. In contrast, if the online machines are slightly faster than the offline machines, it is possible to be constant competitive with regard to the number of machines needed.

Resource augmentation In ESA 2016, Lucarelli et al. [24] consider the problem of minimizing the total weighted flow time of jobs on unrelated machines where preemption is not allowed. This problem is completely hopeless from an online perspective: even on a single machine, there is a lower bound of $\Omega(\sqrt{n})$, and no online algorithm with a performance guarantee is known.

However, the authors show that if you combine two kinds of resource augmentation, you can do something even in this model. Specifically, you have to be able to reject some jobs, and you need faster machines than the offline algorithm. This may seem like it makes the problem potentially very easy for the online algorithm, but in fact the online algorithm only needs to be able to reject some $\varepsilon_r > 0$ fraction (by weight) of the jobs and have machines that are $1 + \varepsilon_s$ as fast as the offline machines, for some $\varepsilon_s > 0$. This is already enough to achieve a competitive ratio of $O(1/(\varepsilon_s\varepsilon_r)$. It is remarkable that a constant competitive ratio can be achieved using arbitrarily little resource augmentation, albeit of two types.

Parallelizable jobs In SODA 2016, Agrawal et al. [1] consider the problem of minimizing the total (or average) flow time for parallelizable jobs that are represented as directed acyclic graphs. A scalable algorithm is one that achieves a competitive ratio of $O(f(\varepsilon))$ for some function f using a speedup of $1 + \varepsilon$. Agrawal et al. [1] show that the well-known LAPS algorithm is scalable and achieves a competitive ratio of $O(1/\varepsilon^3)$.

The LAPS algorithm splits the processing power equally among the ε fraction of the jobs which arrived last. Hence, it needs to know ε in order to run, and it is unclear how you would select this value in practice. Moreover, giving a set of jobs equal processing time is hard to achieve in practice with low overheads.

The authors therefore also consider a more practical algorithm, namely Shortest Job First (SJF). This very simple and natural algorithm sorts jobs by original size (i.e., it ignores processing that has been done on jobs) and gives the highest priority to the smallest jobs. They show that this algorithm achieves only slightly worse results than LAPS, namely a competitive ratio of $O(1/\varepsilon^4)$ at a speedup of $2+\varepsilon$. This is the first greedy algorithm which is shown to perform well for parallelizable jobs in the online setting.

Agrawal et al. [2] consider this scheduling problem further in SPAA 2016, this time with the goal of minimizing the *maximum* flow time. They show that First In First Out is scalable with competitive ratio $O(1/\varepsilon)$. They also consider a work stealing algorithm called Admit First which is shown to have a maximum flow time of $O(\max\{OPT, \log n\}/\varepsilon^2)$ at speed $1+\varepsilon$ with high probability. Work stealing is a practical and efficient scheduler that is used in many parallel languages and libraries.

For the version of the problem where the jobs have weight, the authors show that the algorithm Biggest Weight First is scalable with a competitive ratio of $O(1/\varepsilon^2)$, which is optimal.

4 Various problems

Regarding online bin packing in one or more dimensions, let me just refer you to a new survey [14] which has just come out and which gives a comprehensive overview of results up to and including 2016 (with references to one-dimensional results as well, despite the title).

In ESA 2016, there was a paper on online bipartite matching with advice [18], which has already been discussed in a previous issue [9]. Let me just mention here one of their results, to show that this type of research can also lead to new results without advice: they show that using cn random (not advice) bits, a competitive ratio which approaches $1 - 1/e$ very quickly as c increases can be achieved. This contrasts to the previous result by Karp, Vazirani and Vazirani in STOC 1990 [23] which used $\Theta(n \log n)$ bits to achieve (exactly) $1 - 1/e$.

Online pricing with impatient bidders In SODA 2016, Marek Cygan et al. [15] consider the problem of setting prices for impatient bidders. There is an unlimited supply of identical items, and bidders arrive and leave online. Impatient means that each bidder will buy an item as soon as it is within their budget and if it is within their time interval. The goal is to maximize the total revenue.

Somewhat surprisingly, it turns out to be irrelevant whether the auctioneer knows the deadline of each bidder or not. With or without this knowledge, a competitive ratio of $\Theta(\log h/ \log\log h)$ can be achieved deterministically, and a ratio of $\Theta(\log\log h)$ can be achieved using randomization. Here it is assumed that all budgets of bidders are in the range $[1, h]$.

The way that these results are achieved is by reducing the problem to Veblen bidders. These are bidders which will only buy an item if it reaches their desired price level exactly. If the item becomes too cheap, they lose interest in the good! The authors note that there are many example of Veblen goods, which are goods that people no longer buy if the price becomes too low. They give smartphones as an example for some countries.

Veblen bidders are conceptually easier to handle, but the results that can be achieved with them are tightly related to those for impatient bidders: you only lose a logarithmic factor when going from Veblen bidders to impatient bidders. The paper gives a deterministic tight bound of $\Theta(k/ \log k)$ for Veblen bidders with k price levels, which translates to the result claimed above. This algorithm makes a tradeoff between minimizing the revenue lost from bidders which lose interest and maximizing the revenue in the current round.

For randomized algorithms, they give a tight bound of $\Theta(\log k)$, which is achieved by reducing the problem to the case where no new bidders arrive. This reduction requires randomized guessing which increases the competitive ratio by a factor of $O(\log k)$, but the resulting problem can then be solved with constant competitive ratio.

Online contention resolution schemes Finally, in the fifth paper on online algorithms in SODA 2016, Moran Feldman et al. [20] introduced the concept of online contention resolution schemes (OCRS), which have applications in many online selection problems, including Bayesian online selection, oblivious posted pricing mechanisms, and stochastic probing models. Their framework allows them to formulate the first constant-factor constrained oblivious posted price mechanism for matroid constraints, and the first constant factor algorithm for weighted stochastic probing with deadlines. Additionally, they give optimal guarantees, up to constant factors, for a class of online submodular function maximization problems.

Multi-level aggregation Consider requests which arrive at the nodes of an edge-weighted tree T. Requests are served by repeatedly selecting subtrees from T; the cost of one such service is the total weight of the selected subtree. The goal is to minimize the total waiting costs of all requests plus the total cost of all service subtrees. This problem (MLAP) generalizes the well-known TCP acknowledgment problem (which corresponds to trees of depth 1) and the joint replenishment problem (trees of depth 2). In ESA 2016, Bienkowski et al. [7] give the first constant competitive online algorithm for trees of more than depth 2, with a competitive ratio of $O(D^4 2^D)$ for trees of depth D. This algorithm works for arbitrary waiting cost functions, including the variant with deadlines.

For both results, the authors use a reduction of the general problem to the special case of trees with fast decreasing weights. The algorithm is based on carefully constructing a sufficiently large service tree whenever it appears that an urgent request must be served. The structure of the service tree is exploited in an amortization argument.

Finally, the authors show that known lower bound techniques cannot be used to show any lower bound above 4, and they consider the special case of the problem where T is a path.

References

[1] Kunal Agrawal, Jing Li, Kefu Lu, and Benjamin Moseley. Scheduling parallel DAG jobs online to minimize average flow time. In *SODA*, pages 176–189. SIAM, 2016.

[2] Kunal Agrawal, Jing Li, Kefu Lu, and Benjamin Moseley. Scheduling parallelizable jobs online to minimize the maximum flow time. In *Proceedings of the 28th ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '16, pages 195–205, New York, NY, USA, 2016. ACM. `doi:10.1145/2935764.2935782`.

[3] Noga Alon, Baruch Awerbuch, Yossi Azar, Niv Buchbinder, and Joseph Naor. The online set cover problem. *SIAM J. Comput.*, 39(2):361–370, 2009.

[4] Nir Andelman, Yishay Mansour, and An Zhu. Competitive queueing policies for qos switches. In *SODA*, pages 761–770. ACM/SIAM, 2003.

[5] Yossi Azar, Umang Bhaskar, Lisa Fleischer, and Debmalya Panigrahi. Online mixed packing and covering. In *SODA*, pages 85–100. SIAM, 2013.

[6] Yossi Azar, Niv Buchbinder, T.-H. Hubert Chan, Shahar Chen, Ilan Reuven Cohen, Anupam Gupta, Zhiyi Huang, Ning Kang, Viswanath Nagarajan, Joseph Naor, and Debmalya Panigrahi. Online algorithms for covering and packing problems with convex objectives. In *FOCS*, pages 148–157. IEEE Computer Society, 2016.

[7] Marcin Bienkowski, Martin Böhm, Jaroslaw Byrka, Marek Chrobak, Christoph Dürr, Lukas Folwarczny, Łukasz eż, Jiří Sgall, Nguyen Kim Thang, and Pavel Veselý. Online Algorithms for Multi-Level Aggregation. In Piotr Sankowski and Christos Zaroliagis, editors, *24th Annual European Symposium on Algorithms (ESA 2016)*, volume 57 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 12:1–12:17, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:http://dx.doi.org/10.4230/LIPIcs.ESA.2016.12`.

[8] Martin Böhm, Marek Chrobak, Łukasz Jeż, Fei Li, Jiří Sgall, and Pavel Veselý. Online packet scheduling with bounded delay and lookahead. In *ISAAC*, volume 64 of *LIPIcs*, pages 21:1–21:13. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[9] Joan Boyar, Lene M. Favrholdt, Christian Kudahl, Kim S. Larsen, and Jesper W. Mikkelsen. Online algorithms with advice: A survey. *SIGACT News*, 47(3):93–129, August 2016. `doi:10.1145/2993749.2993766`.

[10] Niv Buchbinder and Joseph Naor. The design of competitive online algorithms via a primal-dual approach. *Foundations and Trends in Theoretical Computer Science*, 3(2-3):93–263, 2009.

[11] Lin Chen, Nicole Megow, and Kevin Schewior. An o(log m)-competitive algorithm for online machine minimization. In *SODA*, pages 155–163. SIAM, 2016.

[12] Lin Chen, Nicole Megow, and Kevin Schewior. The power of migration in online machine minimization. In *SPAA*, pages 175–184. ACM, 2016.

[13] Francis Y. L. Chin and Stanley P. Y. Fung. Online scheduling with partial job values: Does timesharing or randomization help? *Algorithmica*, 37(3):149–164, 2003.

[14] Henrik I. Christensen, Arindam Khan, Sebastian Pokutta, and Prasad Tetali. Approximation and online algorithms for multidimensional bin packing: A survey. *Computer Science Review*, 2017. doi:10.1016/j.cosrev.2016.12.001.

[15] Marek Cygan, Marcin Mucha, Piotr Sankowski, and Qiang Zhang. Online pricing with impatient bidders. In *SODA*, pages 190–201. SIAM, 2016.

[16] Sina Dehghani, Soheil Ehsani, Mohammad Taghi Hajiaghayi, Vahid Liaghat, Harald Räcke, and Saeed Seddighin. Online weighted degree-bounded Steiner networks via novel online mixed packing/covering. In *ICALP*, volume 55 of *LIPIcs*, pages 42:1–42:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[17] Sina Dehghani, Soheil Ehsani, MohammadTaghi Hajiaghayi, and Vahid Liaghat. Online degree-bounded Steiner network design. In *SODA*, pages 164–175. SIAM, 2016.

[18] Christoph Dürr, Christian Konrad, and Marc P. Renault. On the power of advice and randomization for online bipartite matching. In *ESA*, volume 57 of *LIPIcs*, pages 37:1–37:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[19] Matthias Englert and Matthias Westermann. Considering suppressed packets improves buffer management in QoS switches. In *SODA*, pages 209–218. SIAM, 2007.

[20] Moran Feldman, Ola Svensson, and Rico Zenklusen. Online contention resolution schemes. In *SODA*, pages 1014–1033. SIAM, 2016.

[21] Michael H. Goldwasser. A survey of buffer management policies for packet switches. *SIGACT News*, 41(1):100–128, 2010.

[22] Bruce Hajek. On the competitiveness of on-line scheduling of unit-length packets with hard deadlines in slotted time. In *Conference on Information Sciences and Systems*, pages 434–439, 2001.

[23] Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *STOC*, pages 352–358. ACM, 1990.

[24] Giorgio Lucarelli, Nguyen Kim Thang, Abhinav Srivastav, and Denis Trystram. Online non-preemptive scheduling in a resource augmentation model based on duality. In *ESA*, volume 57 of *LIPIcs*, pages 63:1–63:17. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[25] Akira Matsubayashi. Non-greedy online Steiner trees on outerplanar graphs. In *WAOA*, volume 10138 of *Lecture Notes in Computer Science*, pages 129–141. Springer, 2016.

[26] Cynthia A. Phillips, Clifford Stein, Eric Torng, and Joel Wein. Optimal time-critical scheduling via resource augmentation (extended abstract). In *STOC*, pages 140–149. ACM, 1997.