SIGACT News Online Algorithms Column 32: 2017 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 2017. I plan to make this column an annual feature in the December issue (rather than the March issue of the following year) in order to be in time for Christmas shopping...

As before, quite a number of papers on online algorithms have appeared (ten in SODA alone) and I have made a selection. There seem to have been a remarkable number of papers on lower bounds this year, showing that for quite a number of problems, the best known online algorithm so far is in fact the best possible online algorithm. It is great to see so many problems being resolved, and to see so much interest in online algorithms. 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 matching problems

In FOCS, Nayyar and Raghvendra [28] consider the general metric bipartite matching problem. This is a generalization of the problem of online matching on the line, which was recently considered in this column [30, 31]. There is a set of n servers S in a metric space M. Requests arrive online and must be matched to the servers; there will be n requests in total. Let $\mu_M(S)$ be the maximum ratio of the traveling salesman tour and the diameter of any subset of S. Then any online algorithm has competitive ratio $\Omega(\mu_M(S))$. The authors present an algorithm with competitive ratio $O(\mu_M(S)\log^2 n)$. For the case where S is a set of points on the line, this gives a ratio of $O(\log^2 n)$, substantially improving on $O(n^{0.59})$ by Antoniadis et al. [3]. This holds even if the requests are *not* necessarily on that line.

For the case where S is a set of points spanning a subspace with doubling dimension d, the competitive ratio is $O(n^{1-1/d}\log^2 n)$. The algorithm was already presented in Raghvendra [29]. It works by maintaining an offline matching throughout. For this offline matching, it computes a good way to serve the incoming request in each step. This then guides the decision of which server to use for the online matching. In [29], the same problem had been considered, but without considering specific metric spaces. The results there were that the presented algorithm achieves optimal performance both in the adversarial model (ratio $2n-1+o(1)$, where the lower bound is $2n-1$) and in the random arrival model (ratio $2H_n-1+o(1)$; the author gives a lower bound of $2H_n-1-o(1)$).

There are several models for online matching. Buchbinder et al. [16] consider the edge arrival model in ESA. In this model, edges of an unknown graph appear one by one and must be irrevocably accepted or rejected. In each step, the set of accepted edges must form a matching. The authors present a new family of randomized algorithms and show that an algorithm in this family achieves a ratio of 5/9 when the underlying graph is a forest, while none is better. They also give a general upper bound (hardness result) of 0.5914. This bound also holds for trees of maximum degree at most 3.

Their so-called MinIndex framework maintains k matchings, which are ordered. Each arriving edge is added to the first matching in the fixed ordering to which it can feasibly be added (maintaining a matching). In the end, one of the k matchings is returned as the output, according to a probability distribution which is fixed in advance by the algorithm. The algorithms are analyzed using a primal-dual framework.

2 Matching with delays

In STOC 2016, Emek et al. [22] had introduced min-cost perfect matching with delays. In this problem, requests in a finite metric space M must be matched to eachother (rather than to servers). Serving requests immediately can lead to results that are arbitrarily bad from a competitive analysis point of view (as the title of the paper says, haste makes waste). The authors therefore consider a model in which an algorithm is allowed to delay the matchings. The cost of the algorithm in this model is the sum of the distances between matched requests plus the sum of times each request waited since it arrived until it was matched.

The paper presents a randomized online algorithm with competitive ratio $O(\log^2 n + \log \Delta)$ for this problem. Here n is the number of points in M and Δ is the aspect ratio of M. Note that there can be more than n requests in the input. The analysis is based on the study of two interleaved Poisson processes. The algorithm also works in a setting where unmatched requests can be cleared at a fixed penalty. The authors conjecture that the competitive ratio of any online algorithm must depend on n. Finally, they show that the natural deterministic version of their algorithm is $\Omega(n)$-competitive.

In SODA 2017, Azar et al. [6] present an algorithm with competitive ratio $O(\log n)$, thus improving the ratio and removing the dependence on the aspect ratio, which can be unbounded in n. The algorithm is based on a deterministic algorithm for metric induced by edge-weighted trees of height h. This algorithm has cost at most $O(1)$ times the connection cost (the sum of the distances) plus $O(h)$ times the delay cost of every feasible solution. Its analysis is relatively simple compared to the previous result. To solve the problem for arbitrary metrics, a randomized embedding into hierarchically separated trees of height $O(\log n)$ and distortion $O(\log n)$ is used. They also provide a randomized lower bound of $\Omega(\sqrt{\log n})$, thus confirming the conjecture of Emek et al. [22]. The lower bound was improved to $\Omega(\log n / \log\log n)$ by Ashlagi et al. [4] in APPROX. This paper focuses on the bipartite version, where each request is positive or negative and must be matched to a request of opposite polarity. The authors adapt the algorithm of Emek et al. [22] for this case and provide a simplified analysis to improve the competitive ratio to $O(\log n)$. It is based on an $O(h)$-competitive randomized algorithm on weighted trees of height h.

Finally, in WAOA, Bienkowski et al. [13] provided the first *deterministic* algorithm for this problem, achieving a competitive ratio of $O(m^{2.46})$, thus also avoiding any dependence on the aspect ratio. Here $2m$ is the number of requests (which can be larger than n in case some points are repeated). This algorithm does not need to know the metric space in advance. It works by considering the waiting times of requests. As soon as the total waiting time of a pair of requests exceeds the distance between them (times some factor), and the waiting times are at most a given factor apart, then this pair is matched to eachother.

Azar et al. [9] considered a somewhat related problem in STOC 2017. Here, n points in a metric space issue service requests over time, and there is a server which serves these requests. The goal is to minimize the sum of the distances traveled by the server and (a function of) the total delay in serving the requests. The main result is a competitive ratio $O(\log^4 n)$, achieved by a so-called preemptive service algorithm. They show that it is not enough to balance the service costs and the penalty costs locally. There are examples where requests that have not incurred any penalty at all need to be served. The algorithm therefore balances these costs globally, which is more difficult. The results are also generalized to the case of $k > 1$ servers. Finally, special metrics like uniform and star metrics are considered; stronger results are obtained here.

3 Bin and vector packing

Balogh et al. [11] resolved the problem of cardinality constrained online bin packing in ESA. In this problem, items with varying sizes arriving online need to packed into bins of size 1. Each bin may contain at most k items for some predetermined k, and each bin may contain items of total size at most 1. They provide a lower bound of 2 for this problem (for general k) and also improve the lower bounds for specific values of k substantially. In particular, for large k, the best known lower bound before this paper was only 1.54037, whereas it now tends to 2 relatively quickly. It is rare to see tight results for online bin packing (certainly if the most commonly used *asymptotic* competitive ratio is considered), making this a very nice result.

The lower bound construction works adaptively, by considering how exactly each individual item is packed by an online algorithm. Thus the item sizes are not set in advance. The procedure creates items that will be called small and items that will be called large, any large item is larger than any small item, and there is a requirement on the size ratio that will be satisfied (a multiplicative gap between the size of the smallest large item and the largest small item). In previous works, only additive gaps were created, and this did not work very well to create lower bounds for $k > 2$.

The authors also use this idea to show a lower bound strictly larger than 2 on the asymptotic competitive ratio of the online 2-dimensional vector packing problem (where vectors are packed into 2-dimensional bins), and thus provide for the first time a lower bound larger than 2 on the asymptotic competitive ratio for the vector packing problem in any fixed (constant) dimension. In a followup paper in WAOA [10], the same authors furthermore improve the lower bound for online square packing from 1.68 to 1.75.

In that paper, they also consider two alternative settings for one-dimensional bin packing. In the first setting, the value of the optimal offline solution is given in advance. The lower bound is improved from 1.32 to 1.40. In the second setting , each item has a color as well, and each bin may contain items of at most t colors. For $t = 2$, they improve the lower bound from 1.56 to 1.71, and for $t = 3$, they improve the lower bound from 5/3 (which was proved using items of equal size) to 1.80.

This vector packing problem was also considered by Azar et al. [8] in SODA. They called it the vector bin packing problem to distinguish it from the vector knapsack problem (see below). The authors give a general technique to construct lower bounds using linear programming duality; previously, duality had been used to obtain positive results, e.g., via the primal-dual scheme. In particular, for vector bin packing, they prove a lower bound which tends to e as the dimension d tends to infinity. This result is tight, as it matches a previous upper bound [7].

The authors provide two additional tight lower bounds for other problems. The first is ad auctions, where n bidders, each with their own budget $B(i)$, are bidding on products that arrive online. An algorithm must allocate the items to bidders without exceeding their budgets. The goal is to maximize the total revenue. For the case where the number of buyers interested in each item is bounded by a parameter d, a tight lower bound of $1 - (1 - \frac{1}{d})^d$ is presented.

The last problem considered in [8] is capital investment. In this problem, many units of a particular commodity must be produced. A set of machines is available. Each machine has a production cost p_i and a capital cost c_i. In order to use a machine, it first must be bought (at cost c_i). After that, producing one item on that machine costs p_i. Orders for units arrive online and the goal is to minimize the total cost. For the case where all machines are known at the start, a tight lower bound of e is presented.

Chan et al. [18] considered the vector packing problem, where vectors need to be packed into a single d-dimensional knapsack. It is allowed to dispose of previously accepted items, but such items can then not be added in again later. The goal is to maximize a monotone submodular function of the set of accepted items. The paper focuses on the important parameter k, which is the maximum number of nonzero coordinates of any vector. Every coordinate of every vector in the input is required to be at most $1 - \epsilon$ for some $\epsilon > 0$.

The authors present an $O(k/\epsilon^2)$-competitive algorithm, as well as lower bounds of $\Omega(k)$ (deterministic) and $\Omega(k/\log k)$ (randomized). For the case $\epsilon = 0$, a lower bound of $\Omega(k)$ even for randomized algorithms is given. Finally, even if vectors are required to be arbitrarily small in every dimension, a lower bound of $\Omega(\log k/\log\log k)$ is proved. In contrast, Kesselheim et al. [26] achieved a competitive ratio of $1 + \delta$ for this case in the setting where the arrival order of the items is random. The algorithm works by maintaining a fractional solution, which is rounded at every step to achieve an integer solution. The algorithm considers the densities of the vectors (the value per unit weight) in each dimension separately. When a new item arrives, its accepted fraction is continuously increased up to at most 1. At the same time, for each of its k nonzero dimensions, the fraction of the currently least dense accepted item is decreased. This continues as long as the rate of increase in value due to the new item is at least some factor times the rate of loss due to disposing items fractionally. At this point, the algorithm checks whether the new item was accepted with a fraction larger than some threshold α. In this case, the item will be accepted completely. Any existing item for which its accepted threshold drops below another threshold β is removed from the integer solution.

4 Online scheduling

In SODA, Molinaro [27] considered the classic online load balancing problem. In this problem, jobs arrive online (not over time) and need to be assigned to machines. Each job is assigned to one machine and the goal in this paper is to minimize the ℓ_p norm of the machine loads. Molinaro considers a setting in which each job can be processed in k different ways on each of the m machines. This is a generalization of scheduling unrelated machines to minimize makespan. It has long been known that a greedy algorithm that for each job chooses the processing option that least increases the ℓ_p load has competitive ratio $O(p)$, and this is optimal.

Molinaro extends this result by giving algorithms that are optimal (up to constant factors) for both the standard adversarial model (competitive ratio) and the random-order model (secretary model). The algorithm works by "restarting" at time $n/2$. That is, as soon as $n/2$ jobs have arrived, the algorithm ignores everything that has happened so far and pretends all machines are empty. Of course, this means that it is not an online algorithm anymore, since the algorithm must know n in advance in order for this to work.

In SPAA, Agrawal et al. [1] consider jobs that can be represented as directed acyclic graphs, showing the dependencies of tasks within the jobs: different tasks from the same job can run in parallel, but only if all their predecessors have been completed. In their setting, each job i has an associated nonincreasing profit function $p_i(t)$, which gives the profit for completing this job by time t. They give an $2+\epsilon$-competitive algorithm using $O(1/\epsilon^6)$ resource augmentation on the speed, and also show that a speed of at least $2-1/m$ is needed to be $O(1)$-competitive.

5 Various problems

Set Cover Gupta et al. [24] study the classic set cover problem in STOC. They consider the fully-dynamic version of the problem in whcih items arrive and leave, and the goal is to maintain a competitive solution at all times (with regard to the current optimal solution). Dynamic algorithms have a restriction on the running time for each update, whereas online algorithms have a restriction on the number of updates they are allowed to make to the solution (the recourse).

The authors give a dynamic algorithm which achieves $O(\log n)$-competitiveness with $O(f \log n)$ update time, and an algorithm which achieves $O(f^3)$-competitiveness with $O(f^2)$ update time. Here f is the maximum frequency of any element. The second algorithm is the first deterministic constant competitive constant update time algorithm for fully-dynamic vertex cover.

They also give an online algorithm with competitive ratio $O(\min \log n, f)$ with constant recourse. This matches the best offline bounds with $O(1)$ recourse. The results are based on two algorithmic frameworks, based on the classic greedy and primal-dual frameworks for offline set cover.

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. [12] had given the first constant competitive online algorithm for trees of depth 2, with a competitive ratio of $O(D^4 2^D)$ for trees of depth D.

In SODA 2017, Buchbinder et al. [15] improved this to $O(D)$. The algorithm works by transforming the tree into a forest of 3-decreasing trees. In such a tree, the edge costs go down by a factor of at least 3 on each step of a path from the root to a leaf. The idea is to recursively aggregate requests to be served into a subtree, starting from the root. Requests are selected in such a way that as many requests as possible are aggregated while keeping the cost of the tree bounded in order to relate it to the optimal solution. This is done by providing budgets to selected nodes which can be used to aggregate additional nodes.

Online TSP on the line Bjelde et al. [14] completely resolved the online traveling salesman problem on the line. In this problem, requests appear online over time on the real line and need to be visited by a server which is initially located at the origin. For the closed version of the problem, where the server needs to return to the origin after visiting all requests, the authors provide an algorithm with competitive ratio $(9+\sqrt{17})/8 \approx 1.64$, matching a lower bound by Ausiello et al. [5]. The algorithm uses waiting to avoid certain bad inputs. In particular, the case where there are requests at both sides of the server, but the optimal solution serves them in a different order than the online algorithm, is critical.

For the open version of the problem, a lower bound of 2 is trivial: at time 1, release a request at position 1 or -1, depending on which is furthest away from the online server. The authors improve this to approximately 2.04 (the root of a fourth-degree equation) and show that this is tight by giving a matching upper bound. In this version, the server does not need to return to the origin, but the algorithm presented by the authors returns to the origin whenever possible. The advantage of staying near the origin is that the algorithm can delay the choice in which order to serve the extreme requests to the right and to the left for as long as possible.

Network design In the buy-at-bulk network design problem, there is an underlying graph with edge lengths $d(e)$. Sending x_e flow over an edge e costs $d(e)f(x_e)$, where f is a concave cost function. The goal is to find a routing for demand arriving online in order to minimize the total cost. Previous results for this problem used tree-embeddings, giving randomized algorithms with competitive ratios that depend on the number of nodes in the metric. In SODA, Gupta et al. [25] present a deterministic online algorithm with a competitive ratio logarithmic in k, the number of terminals that have arrived. This matches a known lower bound even for the online Steiner tree problem.

They also consider the case where the function f is not known to the online algorithm, and give a deterministic online algorithm with competitive ratio logarithmic in k. The algorithms work by assigning integer types to terminals, and routing each terminal through a sequence of terminals with increasing types, using a spanner-type construction. The authors construct low-stretch spanners in the online setting. They also extend the notion of light approximate shortest-path trees (LASTs) to multi-sink LASTs, where both sources and sinks arrive over time, and a set of edges needs to be maintained so that every source preserves its distance to the closest sink arriving before it at minimum total cost. As part of their algorith, the authors give a tight deterministic algorithm for MLASTs.

Graph coloring Albers and Schrank [2] considered online graph coloring in ESA. For the general case, this problem had essentially been resolved, as an upper bound of $O(n \log n)$ and a nearly matching lower bound of $\Omega(n/\log^2 n)$ is known, which holds even for randomized algorithms that have lookahead or a buffer of size $O(\log^2 n)$.

For specific graph classes however, better results are possible. For instance for trees, First Fit achieves a competitive ratio of $O(\log n)$, which is optimal. Albers and Schrank consider chordal graphs, in which every induced cycle with four or more vertices has a chord. They give a lower bound of $\Omega(d \cdot \log n)$ for the required number of colors by a randomized online algorithm, where d is the chromatic number of the graph. That is, the competitive ratio is at least $\Omega(\log n)$, and First Fit is optimal. Very few lower bounds for randomized algorithms were previously known. The same results are obtained for inductive, bounded-threewidth, and strongly chordal graphs.

Online house numbering Devanny et al. [19] published a paper on the online house numbering problem in ESA. In this problem, houses are added arbitrarily along a road and need to be assigned labels that are in ascending order along that road. The goal here is to minimize the maximum number of times that any house is relabeled, in contrast to minimizing the number of relabelings per update as in previous papers. If labels may be arbitrarily large, it is possible to avoid relabelings completely by assigning to each new house the average of the labels of the houses to the left and to the right of it. This however requires $\Omega(n)$ bits per label. The authors therefore also aim to minimize the number of bits used in the labels. Letting $h(n)$ be the maximum number of bits per label and $g(n)$ be the maximum number of relabels. Then $h(n) = \Omega(\log n)$ and previous results imply that if $h(n) = O(\log n)$, then $g(n) = \Omega(\log n)$. The authors provide three data structures which achieve the following values for $(h(n), g(n))$: $(O(\log^2 n), O(\log n)), (O(\log n), O(\log^2 n)), (O(n^\epsilon), \log n + O(1/\epsilon))$. They use a tree-like structure in which elements that get relabeled often are promoted nearer to the root of the tree, ensuring fewer relabels in the future.

Differential privacy Differential privacy is a formal guarantee that an algorithm run on a sensitive dataset does not reveal too much about any individual in that dataset. There are three main models for the selection of queries, which is adversarial in all cases. In the offline model, the queries are chosen at the beginning, and answered in one step. In the online model, the queries are still chosen at the beginning, but the adversary determines the order in which they arrive. Finally, in the adaptive model, the queries are chosen one at a time, and may depend on the answers given to previous queries.

Many known differentially private mechanisms work just as well in the adaptive model as in the offline model, whereas most lower bounds are given for the offline setting. In SODA, Bun et al. [17] show separations between these models for the first time. That is, they give a family of queries such that exponentially more queries can be answered in the offline model than in the online model, and another family of queries such that exponentially more queries can be answered in the online model than in the adaptive model (with the same guarantee for differential privacy).

Online unit clustering Let me conclude by mentioning a WAOA paper by Dumitrescu and Tóth [20]. They consider a unit clustering problems, where points in $\mathbb{R}^n$ arrive online and need to be put into clusters. A cluster in this case is a d-dimensional unit hypercube. The location of the hypercube is not set in advance; the only constraint is that it must contain the points assigned to it at all times.

Dumitrescu and Tóth give a lower bound of $\Omega(d)$ for this problem, answering a question that had been asked a few times over the years [23, 21]: does the competitive ratio of this problem grow with the dimension? They also give a randomized $O(d^2)$-competitive algorithm for the unit covering problem, where the location of the hypercube must be specified when it is opened. Previously, only a ratio of $O(2^d d \log d)$ was known.

References

[1] Kunal Agrawal, Jing Li, Kefu Lu, and Benjamin Moseley. Brief announcement: Scheduling parallelizable jobs online to maximize throughput. In *SPAA*, pages 87–89. ACM, 2017.

[2] Susanne Albers and Sebastian Schraink. Tight bounds for online coloring of basic graph classes. In *ESA*, volume 87 of *LIPIcs*, pages 7:1–7:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[3] Antonios Antoniadis, Neal Barcelo, Michael Nugent, Kirk Pruhs, and Michele Scquizzato. A o(n) -competitive deterministic algorithm for online matching on a line. In *WAOA*, volume 8952 of *Lecture Notes in Computer Science*, pages 11–22. Springer, 2014.

[4] Itai Ashlagi, Yossi Azar, Moses Charikar, Ashish Chiplunkar, Ofir Geri, Haim Kaplan, Rahul M. Makhijani, Yuyi Wang, and Roger Wattenhofer. Min-cost bipartite perfect matching with delays. In *APPROX-RANDOM*, volume 81 of *LIPIcs*, pages 1:1–1:20. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[5] Giorgio Ausiello, Esteban Feuerstein, Stefano Leonardi, Leen Stougie, and Maurizio Talamo. Algorithms for the on-line travelling salesman. *Algorithmica*, 29(4):560–581, 2001.

[6] Yossi Azar, Ashish Chiplunkar, and Haim Kaplan. Polylogarithmic bounds on the competitiveness of min-cost perfect matching with delays. In *SODA*, pages 1051–1061. SIAM, 2017.

[7] Yossi Azar, Ilan Reuven Cohen, Amos Fiat, and Alan Roytman. Packing small vectors. In *SODA*, pages 1511–1525. SIAM, 2016.

[8] Yossi Azar, Ilan Reuven Cohen, and Alan Roytman. Online lower bounds via duality. In *SODA*, pages 1038–1050. SIAM, 2017.

[9] Yossi Azar, Arun Ganesh, Rong Ge, and Debmalya Panigrahi. Online service with delay. In *STOC*, pages 551–563. ACM, 2017.

[10] János Balogh, József Békési, György Dósa, Leah Epstein, and Asaf Levin. Lower bounds for several online variants of bin packing. *CoRR*, abs/1708.03228, 2017.

[11] János Balogh, József Békési, György Dósa, Leah Epstein, and Asaf Levin. Online bin packing with cardinality constraints resolved. In *ESA*, volume 87 of *LIPIcs*, pages 10:1–10:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[12] Marcin Bienkowski, Martin Böhm, Jaroslaw Byrka, Marek Chrobak, Christoph Dürr, Lukáš Folwarczný, Lukasz Jez, Jiri Sgall, Nguyen Kim Thang, and Pavel Veselý. Online algorithms for multi-level aggregation. In *ESA*, volume 57 of *LIPIcs*, pages 12:1–12:17. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[13] Marcin Bienkowski, Artur Kraska, and Pawel Schmidt. A match in time saves nine: Deterministic online matching with delays. *CoRR*, abs/1704.06980, 2017.

[14] Antje Bjelde, Yann Disser, Jan Hackfeld, Christoph Hansknecht, Maarten Lipmann, Julie Meißner, Kevin Schewior, Miriam Schlöter, and Leen Stougie. Tight bounds for online TSP on the line. In *SODA*, pages 994–1005. SIAM, 2017.

[15] Niv Buchbinder, Moran Feldman, Joseph (Seffi) Naor, and Ohad Talmon. O(depth)-competitive algorithm for online multi-level aggregation. In *SODA*, pages 1235–1244. SIAM, 2017.

[16] Niv Buchbinder, Danny Segev, and Yevgeny Tkach. Online algorithms for maximum cardinality matching with edge arrivals. In *ESA*, volume 87 of *LIPIcs*, pages 22:1–22:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[17] Mark Bun, Thomas Steinke, and Jonathan Ullman. Make up your mind: The price of online queries in differential privacy. In *SODA*, pages 1306–1325. SIAM, 2017.

[18] T.-H. Hubert Chan, Shaofeng H.-C. Jiang, Zhihao Gavin Tang, and Xiaowei Wu. Online submodular maximization problem with vector packing constraint. In *ESA*, volume 87 of *LIPIcs*, pages 24:1–24:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[19] William E. Devanny, Jeremy T. Fineman, Michael T. Goodrich, and Tsvi Kopelowitz. The online house numbering problem: Min-max online list labeling. In *ESA*, volume 87 of *LIPIcs*, pages 33:1–33:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[20] Adrian Dumitrescu and Csaba D. Tóth. Online unit clustering in higher dimensions. *CoRR*, abs/1708.02662, 2017.

[21] Martin R. Ehmsen and Kim S. Larsen. Better bounds on online unit clustering. *Theor. Comput. Sci.*, 500:1–24, 2013.

[22] Yuval Emek, Shay Kutten, and Roger Wattenhofer. Online matching: haste makes waste! In *STOC*, pages 333–344. ACM, 2016.

[23] Leah Epstein and Rob van Stee. On the online unit clustering problem. *ACM Trans. Algorithms*, 7(1):7:1–7:18, 2010.

[24] Anupam Gupta, Ravishankar Krishnaswamy, Amit Kumar, and Debmalya Panigrahi. Online and dynamic algorithms for set cover. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 537–550, 2017.

[25] Anupam Gupta, R. Ravi, Kunal Talwar, and Seeun William Umboh. LAST but not least: Online spanners for buy-at-bulk. In *SODA*, pages 589–599. SIAM, 2017.

[26] Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. Primal beats dual on online packing lps in the random-order model. In *STOC*, pages 303–312. ACM, 2014.

[27] Marco Molinaro. Online and random-order load balancing simultaneously. In *SODA*, pages 1638–1650. SIAM, 2017.

[28] Krati Nayyar and Sharath Raghvendra. An input sensitive online algorithm for the metric bipartite matching problem. In *FOCS*, 2017.

[29] Sharath Raghvendra. A robust and optimal online algorithm for minimum metric bipartite matching. In *APPROX-RANDOM*, volume 60 of *LIPIcs*, pages 18:1–18:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[30] Rob van Stee. SIGACT news online algorithms column 27: Online matching on the line, part 1. *SIGACT News*, 47(1):99–110, 2016.

[31] Rob van Stee. SIGACT news online algorithms column 28: Online matching on the line, part 2. *SIGACT News*, 47(2):40–51, 2016.