SIGACT News Online Algorithms Column 36: 2020 in review

Felix Höhne Sören Schmitt Rob van Stee
University of Siegen, Germany

In this column, we will discuss some papers in online algorithms that appeared in 2020. As usual, we make no claim at complete coverage here, and have instead made a selection. If we 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 us!

1 Online matching

The famous online matching problem continues to inspire new research. In this year's FOCS, Zhiyi Huang et al. [40] improve results for fully online matching, recently introduced by them [39, 38] (see also previous surveys in this series). In this model, all vertices are online. Once a vertex arrives, its incident edges to previously arrived vertices are revealed. This is called an edge-arrival model. Each vertex has a deadline that is after all its neighbors' arrivals. If a vertex is unmatched until its deadline, it must then irrevocably either be matched to another unmatched vertex or be left unmatched. Huang et al. [39] showed that for bipartite graphs the competitive ratio of Ranking is between 0.5541 and 0.5671.

In SODA 2019, Zhiyi Huang et al. [38] (an overlapping set of authors) improved this result to a tight bound of 0.5671. If fractional matchings are allowed then the algorithm water filling, which at each vertex's deadline matches its unmatched portion fractionally to all neighbors with smallest matched portion, achieves a tight competitive ratio of $2 - \sqrt{2} = 0.585$.

In their FOCS 2020 paper, ideas from Water Filling and Ranking are used to give a 0.569-competitive algorithm for integral matching using the online primal dual framework. Essentially, the authors find a way to (in some cases) correct the decisions by Ranking using the decisions by Water Filling (even though that is an algorithm for fractional matching). They also give an algorithm for fractional matching on general graphs called Eager Water-Filling which sometimes matches vertices immediately upon arrival instead of always waiting until the deadline. This algorithm achieves a competitive ratio of 0.592.

In STOC 2020, Zhiyi Huang and Qiankun Zhang consider online matching with stochastic rewards. Each edge has a success probability and the matching along this edge succeeds with that probability. This models online advertising, where it is not certain that a user will click on an ad. This model was introduced by Mehta and Panigrahi in FOCS 2012 [52], who focused on the case where every edge has the same success probability p. They gave results for general p and for the case where p tends to 0 (vanishing probabilities).

Huang and Zhang apply the randomized online primal dual framework using the configuration LP, and achieve a competitive ratio of 0.572 (resp., 0.576) for the case of vanishing and unequal (resp., unequal) probabilities. The previous best results were 0.534 due to Mehta et al. [53] and 0.567 [52].

In another FOCS paper, Matthew Fahbrach et al. [31] consider the less studied edge-weighted version of the online bipartite matching problem. A simple example with three nodes and two edges, where the first arriving edge has weight 1 and the second has weight either 0 or W for some large value W shows that it is not possible to be competitive in this model without additional assumptions. Fahrbach et al. consider the free disposal model, in which previously matched edges may be disposed of for free, a widely-accepted assumption in display advertising. For this model, the authors give the first algorithm which beats the ratio of 1/2 by providing an algorithm with competitive ratio 0.5086.

An important idea in the algorithm is to use a subroutine which the authors call online correlated selection. It ensures that decisions across different pairs of vertices are negatively correlated and ultimately guarantees that a vertex appearing in k pairs is selected at least once with probability greater than $1-2^{-k}$. Essentially, for neighbors of an online vertex, we select the least matched ones. For the vertex-weighted problem, this would mean those vertices that have appeared in the least number of pairs. For the edge-weighted problem, the authors use the online primal-dual framework of Devanur et al. [28] and a formulation of the problem due to Devanur et al. [27] to design a "least-matched" notion.

In APPROX 2020, Nicole Megow and Lukas Nölke [51] considered the problem of online matching on the line, which we have covered extensively in previous editions of this column [59, 60]. All known deterministic algorithms are $\Omega(\log n)$-competitive [3], and the algorithm by Nayyar and Raghvendra [54] has been shown to be $O(\log n)$-competitive by Raghvendra [56]. Megow and Nölke consider the situation where previous matching decisions may be changed to some extent (the algorithm has *recourse*) and they give an $O(1)$-competitive algorithm using $O(\log n)$ recourse. The algorithm interpolates between an $O(\log n)$-competitive online solution without recourse and an $O(1)$-approximate offline solution, i.e., with very large recourse. They also consider the special case of alternating instances, in which no more than one request arrives between two adjacent servers, and give a $(1+\varepsilon)$-competitive algorithm that reassigns each request $O(\varepsilon^{-1.001})$ times. For this special case, the lower bound of $\Omega(\log n)$ without recourse still holds.

In the same workshop, Varun Gupta et al. [37] also give an $O(\log k)$-competitive algorithm for online matching in general metrics, using $O(\log k)$ recourse per client, as well as a 3-competitive algorithm on the line using $O(\log k)$ amortized recourse. They also consider the dynamic setting in which clients and servers may arrive or depart. For this model, they present a simple randomized $O(\log n)$-competitive algorithm with $O(\log \Delta)$ recourse, where Δ is the aspect ratio of the underlying metric. Without recourse, it is not possible to achieve any nontrivial result in this setting.

2 Covering problems

The classic NP-hard set cover problem has been studied extensively. One is given a set X of n elements (universe) and a collection S of m sets, where each set of this collection is associated with a non-negative cost. The goal is to find a cover of minimum cost, that is, a subcollection of S which contains all items in X. Various online versions have also been studied.

In ESA 2020, Yossi Azar et al. [5] consider the online version with delay (SCD). The universe and the family of sets are known in advance, but the requests arrive over time and accumulate delay cost until served by the algorithm. By buying (choosing) a set, the algorithm serves all pending requests on elements of that set. Requests that have not been revealed up to this point in time when the set is bought are not served, hence an algorithm may need to buy a set multiple times. The goal is to minimize the sum of the total buying cost and the total delay cost.

There are two variants of SCD. One is the clairvoyant variant, where the delay function of a request is revealed to the algorithm upon the release of the request. The other one is the non-clairvoyant variant, where at any point in time the algorithm only has knowledge of the delay accumulated up to that point in time. Previous work on this topic [21] provided an $O(\log N)$-competitive clairvoyant algorithm, where N is the number of requests. Because this algorithm relies on oracle calls which compute optimal solutions for NP-hard problems, it has exponential running time.

Yossi Azar et al. present an $O(\log n \log m)$-competitive, randomized, non-clairvoyant algorithm. This is the first known online algorithm for SCD that runs in polynomial time. The main concept is to solve the fractional relaxation problem and then use randomized rounding. Additionally, they prove lower bounds of $\Omega(\sqrt{\log m})$ and $\Omega(\sqrt{\log n})$ on the competitiveness for clairvoyant SCD. This is quite surprising, because it shows that clairvoyance cannot improve competitiveness beyond a quadratic factor. To round out their paper, they present a short and simple deterministic 3-competitive algorithm for vertex cover with delay (a special case of SCD).

Dimitris Fotakis et al. [33] in ICALP 2020 initiate the study of the online Min-Sum Set Cover problem (MSSC). Sets arrive online and an algorithm has to construct a permutation π of elements of X. To serve a request S_t, the algorithm has to pay access cost $\pi_t(S_t)$, which is the position of the first element of S_t in π_t. After every requests at time t the algorithm can change its permutation by paying moving cost $d_{\mathrm{KT}}(\pi_t, \pi_{t+1})$ (known as the Kendall tau distance), that is equal to the number of inversions between π_t and π_{t+1}. The goal is to minimize the overall cost $\sum_t (\pi_t(S_t) + d_{\mathrm{KT}}(\pi_t, \pi_{t+1}))$. The first part of their paper focuses on the static MSSC in the r-uniform case, where all request sets have the same size $|S_t| = r$. Here the online algorithm is compared to a static adversary, that serves the entire request sequence using one permutation. They show a lower bound of $(r+1)(1 - \frac{r}{n+1})$ on the competitiveness of any deterministic online algorithm for the online static MSSC. Further they complement this lower bound by giving a $(5r+2)$-competitive deterministic algorithm called Lazy-Rounding.

Their algorithm is essentially a lazy derandomization of the well-known multiplicative weight updates (MWU) algorithm. With an eye on computational efficiency, the authors remark that Lazy-Rounding is computationally inefficient since it simulates the MWU algorithm, which maintains a probability distribution over all $n!$ permutations. For this reason they propose a memoryless algorithm Move-All-Equally (MAE), which moves all elements of the request set S_t at the same speed towards the beginning of the permutation until the first element reaches the first position. Their analysis shows that the competitive ratio of MAE is $\Omega(r^2)$ and the authors conjecture that an $O(r)$ guarantee cannot be achieved by a memoryless algorithm. Additionally, they show an upper bound of $2^{O(\sqrt{\log n \log r})}$ on the competitiveness of MAE.

The second part of their paper deals with the dynamic variant, where online algorithms are compared against an optimal solution that is allowed to change its permutations over time. In this setting they show that MAE is still $O(r^{3/2}\sqrt{n})$-competitive. By crafting an adversarial instance they further show that for any $r \geq 3$ the competitive ratio of MAE is $\Omega(r\sqrt{n})$.

In SODA 2020, Anupam Gupta and Roie Levin [36] consider the online version of a very general covering problem, the so-called online submodular cover problem. Over time a time-monotone sequence of monotone submodular functions $f^{(1)}, f^{(2)}, \ldots$ is revealed, where time-monotonicity means that any submodular cover for $f^{(t)}$ is also a cover for $f^{(t-1)}$. At each time step t an algorithm must output a set S_t such that $f^{(t)}(S_t) = f^{(t)}(X)$. Additionally, past decisions cannot be revoked, it is required that $S_{t-1} \subseteq S_t$. The goal is to minimize the cost of the output solution. The authors approach this problem by first solving fractional relaxations of covering LPs online and then use randomized rounding to output integer solutions. This results in an efficient randomized algorithm, which guarantees that for each N the expected cost of solution S_N is within $O(\ln n \ln(N \cdot f_{\max}/f_{\min}))$ of the optimal solution, where $f_{\max}$ and $f_{\min}$ are the largest and smallest marginals for any of the functions in the input sequence respectively.

3 The random-order model

When analyzing the competitiveness of an algorithm we usually have an adversary decide the complete input. In some cases this model is too pessimistic and no algorithm can outperform trivial ones. This has prompted the study of beyond worst case models like the random-order model in which an adversary decides the items but they arrive in random order. This model has been receiving increasing attention.

In ICALP 2020, Susanne Albers and Maximilian Janke [1] consider the fundamental makespan minimization problem in this model. They give an algorithm that achieves a competitive ratio of 1.8478, not just in expectation but with high probability. The algorithm is analyzed by identifying a set of properties that a random permutation of the input jobs satisfies with high probability. They also give a lower bound of 4/3 for the expected competitive ratio and show that no algorithm can be better than 3/2-competitive with high probability.

In many real-world scenarios, we do not only have the online input available but also data from the past or other sources. Motivated by this, in SODA 2020 Haim Kaplan et al. [43] extend the standard online worst-case model as well as the random-order model by revealing a random sample of the adversarial input in advance. They study this model on the secretary problem. In the worst case model with sample, the adversary picks $n + h$ candidates and h candidates are revealed to the algorithm for the purpose of learning. They give simple algorithms with a competitive ratio that is essentially optimal. For $h \geq n - 1$ this ratio is $\frac{1}{2}$ with an upper bound of $\frac{1}{2} \cdot \frac{2^n}{2^n - 1}$ for $h \geq n$.

In the random order model with sample, the algorithm functions just like the algorithm for the ordinary secretary problem. The h revealed candidates are used to shorten or remove the sampling phase. When given a sample of size n the algorithm starts making decisions based on random subsets of the observed input. This algorithm is nearly optimal if h is small compared to n but there is an interesting gap for large h.

Finally, the authors show that an algorithm can not at the same time be optimal in the worst case and in the random order model. More precisely, a c-competitive algorithm for the worst case model with sample is at most $(1 - c)$-competitive in the random-order model with sample.

Of interest are also algorithms that combine the best of both worlds and perform well on purely stochastic and purely adversarial inputs. And even more interesting are algorithms that can perform well when given an input that is a mix of both. In order to better understand these kinds of robust algorithms, Thomas Kesselheim et al. in ICALP 2020 [44] propose the *Bursty Adversary plus*

Random Order (BARO) model in which bursts of adversarial time steps are mixed into a random-order input. They study this model on the knapsack secretary problem and give an algorithm that is $1 - \tilde{O}(\frac{\Gamma}{k})$-competitive when the adversarial time steps can be covered by $\Gamma \geq \sqrt{k}$ intervals of size $\tilde{O}(\frac{n}{k})$. (The notation $\tilde{O}$ hides log factors.) Setting $\Gamma = \sqrt{k}$ the algorithm is $(1 - O(\frac{\ln^2 k}{\sqrt{k}}))$-competitive with a fraction of up to $O(\frac{\ln k}{\sqrt{k}})$ of items being adversarial. For large k this is almost optimal, even in the absence of adversarial items.

4 Chasing convex bodies

In the convex body chasing problem, an online algorithm receives a request sequence of convex sets (bodies) $\{K_1, K_2, \dots\} \in \mathbb{R}^d$ and has to respond with a sequence of points $\{x_1, x_2, \dots\}$ located in these sets. The goal is to minimize the total distance between these consecutive points.

In last year's STOC, Sébastien Bubeck et al. [18] were able to prove that there exists an online algorithm with *finite* competitive ratio for the family of convex sets in the Euclidean space. Their presented algorithm is $2^{O(d)}$-competitive. This year's focus has been on improving the competitive ratio.

In SODA 2020, two independent papers present algorithms that are d-competitive for arbitrary normed spaces and $O(\sqrt{d \log T})$-competitive in the Euclidean space, where T is the length of the request sequence. In [58], Mark Sellke explains that chasing convex bodies is a special case of the functional variant called chasing convex functions. For a convex function he defines the functional Steiner point. The main idea of his algorithm is to stay at the functional Steiner point of the work function. The work function $W_t(x)$ is defined as the smallest cost OPT could have at time t while starting at $x_0 = 0$ and ending up at $x_t = x$. As one of his last steps he shows that the functional Steiner point of the work function at time t is indeed an element of K_t.

C.J. Argue et al. [4] propose an algorithm that uses an outer guess-and-double step, which maintains a current estimate r that lies in the interval $[\text{OPT}/2, \text{OPT}]$. At each time step t, their algorithm moves to the Steiner point of the $2r$-level set Ω_t of the work function.

The above mentioned paper by Mark Sellke is an extension to the work of him with Sébastien Bubeck et al. [17]—also published in SODA 2020. They started by investigating the *nested* version of convex body chasing. Here the requested sets have the additional property that each set is contained within the previous one (i.e., they are nested). In the first part of their paper, they analyze the algorithm that moves to the Steiner point of the newly requested set. They prove its competitive ratio to be $O(\min(d, \sqrt{d \log T}))$ which is nearly optimal for sub-exponentially many requests. Further, by comparing the competitive ratio against the Hausdorff distance between K_1 and K_T, they show that the memoryless Steiner point approach actually achieves exact optimal competitiveness for any pair (d, T).

In the second part of their paper, they give a different algorithm, which is nearly optimal even for exponentially many requests. Instead of moving to the Steiner point of K_t, they add a small ball B_r to K_t and select the centroid with respect to a log-concave measure that depends on the normed space. This new approach yields $O(\sqrt{d \log d})$ competitiveness for Euclidean spaces. For normed ℓ^p spaces with $p \geq 1$ this new algorithm is optimal up to a factor of $O(\log d)$.

5 Caching

In the online caching problem, a sequence of items arrives one by one and we have a cache of fixed size. The goal is to minimize the number of page faults. For the traditional online model, there is a lower bound of $\Omega(\log k)$ on the competitive ratio of randomized algorithms.

We can get around this lower bound by changing the model and giving the algorithm access to more information. One possibility is giving the online algorithm access to machine learned advice. This model has previously been applied to a variety of online problems including ski-rental, job scheduling and others. In the case of online caching, an oracle predicts when an element will arrive again. Lykouris and Vassilvitskii [50] showed that there is a prediction augmented caching algorithm with a competitive ratio of $O(1 + \min(\sqrt{\eta/\textsc{Opt}}, \log k))$ where the prediction error is bounded by η. The dependence on k in the competitive ratio is optimal but the dependence on $\eta/\textsc{Opt}$ left room for improvement. Dhruv Rohatgi in SODA 2020 [57] makes progress in closing this gap by providing an improved algorithm with competitive ratio $O(1+\min(\frac{\log k}{k}\frac{\eta}{\textsc{Opt}}, \log k))$ and a lower bound of $\Omega(\log \min(\frac{\eta/\textsc{Opt}}{k \log k}, k))$.

The algorithm in [50] is a marking algorithm that considers eviction chains and trusts the oracle until the chain becomes too long and then ignores the oracle. A marking algorithm works in phases that start with all elements being unmarked. When a cache hit occurs the corresponding element is marked. When a cache miss occurs, some unmarked element is evicted and the new element is inserted and marked. If all elements in the cache are marked and a cache miss occurs, all elements are unmarked and a new phase begins. Rohatgi shows that the simple improvement of only trusting the oracle once at the start of each eviction chain already leads to an improved ratio of $O(1 + \min(\frac{\log \eta}{\textsc{Opt}}, \log k))$. Further improvement of the algorithm requires deviating from the marking based framework by sometimes evicting marked elements.

Alexander Wei [61] in APPROX 2020 directly follows up on this work. They give a substantially simpler algorithm that gives an improved competitive ratio of $O(1 + \min(\frac{1}{k}\frac{\eta}{\textsc{Opt}}, \log k))$. This algorithm compares LRU and BlindOracle, the algorithm that blindly follows the oracle, and follows the one that has performed better so far. This algorithm is optimal among deterministic algorithm, which means randomization is needed to approach Ruhatgi's lower bound.

Zhihao Jiang et al. in ICALP 2020 [42] continue this line of work by considering *weighted* paging (caching) with predictions. As shown by Rohatgi, predicting the next arrival of an element with perfect accuracy gives a competitive ratio of $O(1)$ for unweighted paging. Furthermore it is also possible to obtain a constant approximation when given a lookahead of $l \geq k - 2$ distinct pages. However, neither of these models are sufficient to beat the existing lower bounds for online weighted caching, which are $\Omega(k)$ for deterministic and $\Omega(\log k)$ for randomized algorithms. Jiang et al. show that combining both models allows for a 2-competitive algorithm. In the combined model, when an item p arrives the algorithm is given the next time-step when p will arrive again and additionally all page requests until that request.

Ravi Kumar et al. in SODA 2020 [47] consider a different way of giving the algorithm information about the request sequence. They consider a semi-online model where request sequences are generated by walks on a directed graph called the access graph. The algorithm knows the access graph but not the actual request sequence. Kumar et al. extend this model by creating several individual request sequences using access graphs. The final sequence given to the algorithm is then obtained by interleaving the individual sequences. This approach is motivated by multitasking systems that support a number of tasks at the same time with a limited cache.

For the case of a single access graph, they give a tight competitive ratio of $\Theta(b)$ where b is a structural parameter of the graph called branching depth. For the interleaved model, they give a tight competitive ratio of $\Theta(b + \log t)$ where t is the maximum number of interleaved access graphs. For the case of interleaved cashing, they reduce the problem to a coin hunting game where a hider hides a coin in one of t bins and a seeker, that only knows the probability distribution used by the hider, tries to find that coin. Interestingly, they show that this coin hunting game has other applications besides the interleaved caching model. In particular, it can be used to yield an alternative analysis of the classical random marking algorithm for the standard caching problem.

6 Online scheduling

In SODA 2020, Silvio Lattanzi et al. [49] consider online scheduling with a similar approach as [57, 42] for caching. They use machine learned predictions to overcome lower bounds on the competitive ratio in the classic online problem. Specifically they consider makespan minimization under restricted assignment. This is a special case of unrelated machines. Jobs arrive one at a time, each annotated with its size and a subset of m machines it can run on.

In the classical online setting, there is an $\Omega(\log m)$ lower bound for the competitive ratio. Lattanzi et al. assign weights to each machine, and fractionally assign jobs according to these weights. Given predictions of these weights with error η, they construct a $O(\log \eta)$-competitive fractional assignment. The second step is an online rounding algorithm that rounds any fractional assignment into an integral schedule, losing an $O((\log \log m)^3)$ factor in the makespan. This is nearly optimal, since they also give an $\Omega(\log \log m)$ lower bound for online rounding algorithms. Combining both steps yields a competitive ratio of $O(\log \eta (\log \log m)^3)$.

In ICALP 2020, Cohen et al. [25] consider online 2D load balancing. Two-dimensional vectors arrive and need to be assigned to two-dimensional machines. The goal is to minimize the makespan, which is defined as the maximum load of any machine in any dimension. (The load of a machine in dimension i is the sum of the i-th entries of the vectors assigned to this machine.) This can of course be seen as a special case of vector scheduling, which has already been (asymptotically) resolved. However, it is also a generalization of the widely studied online load balancing problem. The authors argue that this is an important case to study as it models the case where resources of two types are available and we try to minimize their usage.

First, they show that (a straightforward generalization of) the classical load balancing algorithm has competitive ratio exactly 8/3. This is done by analyzing reachable states for priority algorithms [14], which allows us to essentially restrict our attention to two machines. In the analysis, the pair of machines considered needs to be selected carefully, and it is therefore not simply a matter of analyzing an algorithm which only has two machines in the first place.

For the case where OPT is known, the authors give a best fit-type algorithm with competitive ratio exactly 9/4. (For one dimension, the best known result for this problem is 3/2.) This algorithm works by balancing between the two dimensions and minimizing the difference in their loads subject to the constraint that the goal competitive ratio of 2.25 is maintained. They further analyze First Fit and show that it has competitive ratio 2.5 for known OPT and 2.89 for unknown OPT (by maintaining a guess for OPT), beating the simple upper bound of 3 achieved by a greedy algorithm.

Finally, they give an existence result for fixed dimension d, showing that it is (in theory) possible to give a deterministic online algorithm with competitive ratio arbitrarily close to c_d, the optimal competitive ratio against the fractional optimal solution. It is interesting that this can be achieved, especially given that c_d is not actually known. As the authors stress, actually constructing the algorithm (or to be precise, constructing the decision tree that it needs) requires extremely large, though polynomial for fixed d and ε, running time.

In SPAA 2020, Jamalabadi et al. [41] considered an admission control problem on parallel machines. Jobs with deadlines arrive online and need to be assigned to a machine (potentially starting only later) or discarded immediately. The goal is to maximize the total size of the accepted jobs. Of course, the schedule must be such that all accepted jobs complete by their deadlines.

The authors show that if each job has a small amount of slack, it is possible to design a nearly optimal deterministic online algorithm. Here a slack of ε means that for a job with size p, deadline d and release date r, we always have $d \geq (1+\varepsilon)p + r$. The paper first gives a lower bound. The design of the algorithm is then based on this lower bound construction. This is the first such result which holds for parallel machines and does not require job preemption or migration.

7 The k-server problem and metrical task systems

Predrag Krnetić et al. [46] consider the online k-server problem with delays on a uniform metric space (k-OSD). This problem is defined on $k+n$ nodes in a metric space, where the distance between any pair of nodes is 1. An algorithm has access to k servers and over time m requests to locations appear. The algorithm serves a request on a node by moving a server to this node. In the classic k-server problem the algorithm has to serve requests immediately. If instead delays are allowed, the algorithm can wait and incur delay costs instead of answering the request immediately. In the end all requests have to be served. The goal is to minimize the sum of moving and delay costs.

In the clairvoyant setting (where the algorithm knows the delay function of a request when it appears) they use an average technique to prove a lower bound of $2k+1$ on the competitive ratio of all deterministic algorithms. Here they highlight similarities to the classic paging problem.

They then give a construction of an algorithm. This algorithm has a history counter (for each server) that tracks the ordering in which the servers were moved and it is updated with every new request. If a new request appears on a currently occupied node, such a request is served immediately by the server that occupies the node. If instead a new request appears on a node with a hole, an accumulative delay counter is started at this node or the counter is incremented. If this counter reaches the threshold $(k+1)/k$, the algorithm moves a server according to a selection strategy. By an intricate phase partitioning analysis, they show that if this selection strategy fulfills two conditions, the constructed algorithm is $2k+1$-competitive for the k-OSD problem with delays on uniform metrics. The first condition is called *conservativeness*, which makes sure that for every subsequence of requests the servers are not moved too much. The second condition is called *perfect-usefulness*, which makes sure that only one server is used for every critical node (that is a node for which the threshold is reached, but it has not been served yet).

The authors then analyze different well-known page eviction strategies as selection strategies and again highlight the similarities to paging. They show that least recently used (LRU), clock replacement (CLOCK) and first-in/first-out (FIFO) all fulfill the required conditions and hence yield a $2k+1$-competitive algorithm. Even tho flush-when-full (FWF) does not fulfill conservativeness, it still yields a $2k+1$-competitive algorithm.

This shows that clairvoyance does not give any advantage for the k-OSD problem with delays on uniform metrics, because the presented algorithms are designed in the non-clairvoyant setting.

In ISAAC 2020, Sébastien Bubeck and Yuval Rabani [19] consider parametrized versions of metrical task systems and metrical service systems, where the constrained parameter is the number of possible distinct requests m. Formally, a metrical task system instance consists of a finite metric space $\mathcal{M} = (X, d)$, an initial state s_0 and a sequence of requests $\rho_1, \rho_2, \ldots, \rho_L$, where the request ρ_t is a cost function. They denote $n = |X|$. At each time step an online algorithm has to choose a state $s_t \in X$ based only on $\mathcal{M}$ and $\rho_1, \rho_2, \ldots, \rho_t$. The objective is to minimize the sum of the transition costs and the processing costs $\sum_{t=1}^{L}(d(s_{t-1}, s_t) + \rho_t(s_t))$. The authors denote by $c^{\text{det}}_{\mathcal{M},m}$ ($c^{\text{rand}}_{\mathcal{M},m}$, respectively) the best possible competitive ratio that can be achieved by a deterministic (randomized, respectively) online algorithm, when the number of possible distinct requests is restricted by m.

They then analyze how restricting m affects the achievable competitive ratios if $\mathcal{M}$ is a uniform metric space or a paired-uniform metric space. For a uniform metric space they show that for every $m \leq n$, it is $c^{\text{det}}_{\mathcal{M},m} = \Theta(m \log(en/m))$ and $c^{\text{rand}}_{\mathcal{M},2} = \Theta(c^{\text{rand}}_{\mathcal{M}}) = \Theta(\log n)$. This shows that in this setting restricting m helps deterministic algorithms immensely, but not randomized algorithms. If instead $\mathcal{M}$ is a paired-uniform metric space, they show $c^{\text{det}}_{\mathcal{M},2} = \Theta(n)$ and $c^{\text{rand}}_{\mathcal{M},2} = \Theta(\log n)$. Compared to previous results, this show that the restriction does not help the asymptotic performance of deterministic algorithms, nor that of randomized algorithms.

The authors also investigate the impact of restricting m for metrical service systems, that use just two scales, 0 and ∞ (a.k.a. set chasing). They now denote the best deterministic (randomized, respectively) competitive ratio that can be achieved for chasing sets drawn from a collection of m subsets of points in $\mathcal{M}$ by $\hat{c}^{\text{det}}_{\mathcal{M},m}$ ($\hat{c}^{\text{rand}}_{\mathcal{M},m}$, respectively). In this setting the restriction helps both deterministic and randomized algorithms. If $\mathcal{M}$ is a uniform metric space, then for every $m \in \mathbb{N}$, it holds that $\min\{m, n\} - 1 \leq \hat{c}^{\text{det}}_{\mathcal{M},m} \leq \min\{m, n\}$ and $\hat{c}^{\text{rand}}_{\mathcal{M},m} = \Theta(\min\{m, \log n\})$. Their last result is that both the supremums of $\hat{c}^{\text{det}}_{\mathcal{M},m}$, respectively $\hat{c}^{\text{rand}}_{\mathcal{M},m}$, over all L-level hierarchically separated trees $\mathcal{M}$ are between $2^{\lfloor m/2 \rfloor - 1}$ and $m2^m$.

8 Facility location

Another well-studied problem is facility location. In APPROX 2020, Xiangyu Guo et al. [34] considers the models with recourse (see also Megow and Nölke [51] discussed above) and the dynamic version. They give a $(1 + \sqrt{2} + \varepsilon)$-competitive algorithm using $O(\frac{1}{\varepsilon} \log n \log \frac{1}{\varepsilon})$ amortized facility and client recourse, where n is the total number of clients arrived during the process. Together with the randomized local search technique of Charikar and Guha [23], this gives a $(1 + \sqrt{2} + \varepsilon)$-approximate dynamic algorithm with total update time $\tilde{O}(n^2)$ for the setting in which clients only arrive (incremental setting). In the fully dynamic model, they give an $O(1)$-approximation algorithm with $O(|F|)$ preprocessing time and $O(n \log^3 D)$ total update time for HST metric spaces. Here $|F|$ is the number of potential facility locations and D is the diameter of the metric. This also implies an $O(\log |F|)$ approximation with preprocessing time of $O(|F|^2 \log |F|)$ and $O(n \log^3 D)$ total update time [8, 32].

In SPAA 2020, Jannik Castenow et al. [22] consider the multi-commodity version. Here there is a set S of commodities and it needs to be determined which facility will offer which commodities. Each request must be connected to a set of facilities jointly offering the commodities demanded by it. The authors provide an $\Omega(\sqrt{|S|} + \frac{\log n}{\log \log n})$ lower bound for randomized algorithms which already holds on the line. They also give a deterministic $O(\sqrt{|S|}\log n)$-competitive algorithm and a randomized $O(\sqrt{|S|}\frac{\log n}{\log\log n})$-competitive algorithm. Xiangyu Guo et al. [35] in APPROX 2020 consider variants of this problem. In their first result they allow for recourse, meaning the solution can be changed at each step. The recourse per step of an online algorithm is then the number of changes it makes to the solution. They give a $(1+\sqrt{2}+\varepsilon)$ competitive algorithm with $O(\frac{\log n}{\varepsilon}\log\frac{1}{\varepsilon})$ amortized facility and client recourse. n is the total number of clients that arrived during the process.

In the dynamic facility location problem the goal is to maintain a good solution while minimizing the time spend updating the solution. They give a $(1+\sqrt{2}+\varepsilon)$-approximation dynamic algorithm with total update time of $\tilde{O}(n^2)$. This algorithm is based on local search, an algorithm that achieves an approximation factor of $(1+\sqrt{2})$ in the offline setting.

Finally they consider the fully dynamic model, where clients can arrive as well as depart. Their main result is an $O(1)$-approximation algorithm with $O(|F|)$ preprocessing time and $O(n\log^3 D)$ total update time for HST metric spaces. F is the number of potential facility locations and D the diameter of the aspect ratio of the metric space, i.e. the largest distance between two points in it. Using results from [8] and [32] which show that any N-point metric space can be embedded into a distribution over HSTs, such that the expected distortion is at most $O(\log n)$, they obtain a $O(\log|F|)$ approximate algorithm with preprocessing time $O(|F|^2\log|F|)$ and $O(n\log^3 D)$ total update time. This results holds in expectation for every time step in the oblivious adversary model.

9 Various problems

In the best paper of FOCS 2020, Mark Bun et al. [20] investigate the relationship between online learnability and private probably approximately correct (PAC) learnability. In online learning, there is a class of predictors $\mathcal{H} = \{h : X \to \{\pm1\}\}$ over a domain X. Over time an algorithm is fed with examples $(x_1, y_1), \ldots, (x_T, y_T) \in X \times \{\pm1\}$ and at each time step t it has to make a prediction $\hat{y}_t \in \{\pm1\}$. After its prediction, the algorithm learns whether its prediction was correct. The goal is to minimize the regret, which is the difference between the number of mistakes made by the algorithm and the best predictor in $\mathcal{H}$. A class $\mathcal{H}$ is said to be online learnable if for every T, there is an algorithm that achieves sublinear regret $o(T)$ against any sequence of T samples. In PAC learning, the algorithm has access to the whole sample, that is drawn from some unknown distribution and labeled by an unknown hypothesis h^*. The goal is to output a hypothesis $h \in \mathcal{H}$ that mostly matches the labels of h^* on future examples from the unknown example distribution with high probability. A class $\mathcal{H}$ is said to be PAC learnable with complexity n, if n samples are sufficient to obtain error at most α over new examples from the distribution with probability at least $1-\beta$. A differential privacy constraint ensures that changing any one of the samples may not affect the output distributions by the learner by too much.

The authors show that every online learnable class $\mathcal{H}$ is also approximate differentially-privately PAC learnable (i.e. there exists an algorithm for it). This result not only answers an open question of Noga Alon et al. [2] in STOC 2019, it in fact complements the results of Alon et al. and shows the equivalence of online learnability and private PAC learnability.

Also in FOCS, Braverman et al. [16] provide a new framework for randomized algorithms for numerical linear algebra problems in the sliding window model. They also exhibit a connection to the online model, providing online algorithms for several problems and in particular resolving an open question by Bhaskara et al. [12] from the previous FOCS. Finally, they provide another framework for deterministic algorithms, leading to several new online algorithms that use nearly optimal space.

Nikhil Bansal et al. [6] continue their study of minimizing discrepancies. In STOC 2020, they consider the online vector balancing problem. In this problem, T vectors arrive one by one and need to be given a sign. The n coordinates of each vector are taken from some arbitrary distribution over $[-1, 1]$, and the goal is to keep the maximum norm of any signed prefix-sum (the discrepancy) as small as possible. Bansal and Spencer [7] recently gave an $O(\sqrt{n} \log T)$ bound for the case where each coordinate is chosen independently.

In their new paper, they consider the case where there are dependencies. One example of this case is the online interval discrepancy problem. In this problem, T points are sampled one by one uniformly in the unit interval $[0, 1]$, and the goal is to give them signs so that each sub-interval remains nearly balanced. They give an upper bound which is polynomial in n and $\log T$ for this model and a polylog(T) upper bound for online interval discrepancy. To do this, they give new anti-concentration bounds for uncorrelated and pairwise independent variables.

Their work also has applications for the online envy minimization problem, which was introduced by Benade et al. [10] in EC 2018. In this problem, T items arrive one by one and need to be assigned to one of two players, knowing their valuations, and the goal is to minimize cardinal or ordinal envy. Benade et al. gave a lower bound of $\Omega(T^{1/2})$ for the adversarial model. Bansal et al. [6] give an upper bound of $O(\log^3 T)$ for ordinal envy and $O(\log T)$ for cardinal envy when the player valuations are drawn from a distribution.

William Kuszmaul [48] considers the p-processor cup game in SODA 2020. In this game, a filler distributes up to p units of water among n cups. Then an emptier selects p cups and removes up to one unit of water from each. The goal is to minimize the amount of water in the fullest cup which is called the backlog. We can also think of p processors working on one of n tasks each while p new units of work may arrive during each slice of time. So far the positive results on these problem used resource augmentation where the filler is restricted to distributing $(1 - \varepsilon)p$ units of water in each step. Kuszmaul shows that Greedy (the algorithm that always chooses the highest p cups) is $O(\log n)$ competitive and this is optimal for $n \geq 2p$. This is shown using an intricate system of invariants for the p-processor cup game.

Michael Bender et al. [11] proposed an algorithm called smoothed greedy for the variation of the problem with resource augmentation. Kuszmaul shows that this algorithm achieves backlog $O(\log p + \log \log n)$ with probability $1 - 2^{-\text{polylog}(n)}$ for $2^{\text{polylog}(n)}$ steps. For fixed p and n large enough this becomes $O(\log \log n)$ which is asymptotically optimal. This result doubles as a smoothed analysis of the deterministic greedy algorithm.

Also in SODA, Prosenjit Bose et al. [15] propose a new structure for searching elements of a tree-structured space online. Given a graph G, over time a request sequence $X = x_1, x_2, \ldots$ of searches drawn from the vertices of the graph is revealed. Once the pointer from the root has been moved to the requested node, this request counts as served. At unit cost an algorithm can move the pointer from a node to a child or its parent and perform a rotation of the current node. A rotation is swapping two adjacent nodes and reorganizing the search tree accordingly. In the past binary search trees (BSTs) have been the tool to deal with this problem. The authors generalize BSTs to the general search tree model (GST). In the GST a node of the search tree may have multiple children. If the underlying tree G is a path both models are equivalent. They highlight which properties generalize from BST to GST and which do not.

Their main result is an $O(\log\log n)$-competitive algorithm for any search sequence of length $m = \Omega(n)$, where n is the number of vertices of G. This matches the best known competitive ratio for binary search trees. By generalizing the interleave lower bound of BSTs to their model, they are able to give a lower bound on the optimal cost of answering a request sequence. It will be interesting to see how this new model will impact further research on searching on trees.

Probabilistic metric embedding is a tool for designing online algorithms on metric spaces by embedding the underlying metric into a simpler tree metric and solving the problem there. Due to distortion, this incurs some overhead in the competitive ratio. A question is whether this embedding can be constructed online such that the distortion is a polylogarithmic function of k the number of terminals. Yair Bartal et al. [9] in SODA 2020 answer this question negatively by giving a lower bound of $\tilde{\Omega}(\log k \log \Phi)$ where Φ is the aspect ratio of the set of terminals. (In this case, the notation $\tilde{\Omega}$ hides log log factors.) Unfortunately this may result in a polynomial dependence in terms of k, as Φ may be exponential in k. They also show that a careful modification of probabilistic embedding into trees of Bartal [8] in FOCS 1996 has distortion ratio $O(\log k \log \Phi)$ which is therefore optimal up to log log factors.

As a second result, they give a framework that bypasses this lower bound for a broad class of problems which they call abstract network design. For this class they devise an algorithm with only $O(\min(\log k \log(kr), \log^3 k))$ overhead in the competitive ratio. This also implies first polylog(k)-competitive algorithms for the problems subadditive network design and group Steiner forests.

In the online multiple knapsack problem, incoming items have to be either rejected or irrevocably stored in one of n bins. Marcin Bienkowski et al. in ICALP 2020 [13] study the proportional version of the problem where profits and item sizes are equal and the goal is to maximize the sum over all accepted items. The online complexity has been resolved for other variants of this problem. For example if the objective function is to maximize the load in the fullest bin, instead of the sum over the load in all bins, there is an optimal 1/2-competitive deterministic algorithm. The generalization where profits and sizes are unrelated does not admit any competitive algorithms.

For the proportional version, there has been a gap between the 1/2-competitive algorithm FirstFit and an upper bound of $R = 1/(1+\ln 2) = 0.5906$ for randomized algorithms. Bienkowski et al. close this gap by providing a $R - O(1/n)$-competitive deterministic online algorithm that is also optimal up to lower order terms for the class of randomized solutions. They show that the algorithm is indeed optimal and the $O(1/n)$-term is not avoidable in the deterministic case.

The algorithm classifies the items as small, middle or large items, and deals with each class differently. If an algorithm greedily collects large items the adversary may give it n items of size $\frac{1}{2} + \varepsilon$, which are all accepted by Alg, followed by n items of size 1 which are accepted by Opt giving ratio arbitrarily close to $\frac{1}{2}$. An algorithm that stops accepting large items may end up with a very small profit compared to Opt, who keeps accepting those items.

The Rising Threshold algorithm uses an increasing threshold function f to balance these strategies and ensures that the size of the ith accepted item is at least $f(i/n)$. It also carefully manages medium sized items by stacking some of them in the same bin and leaving others in their own bins and available to be combined with larger items.

Boaz Patt-Shamir and Evyatar Yadai introduce and work on a generalization of the well-known ski rental problem in SPAA 2020 [55]. One is confronted with a task of unknown duration (i.e. the stopping time is unknown) and at every time step one has to chose to "buy" or to continue to "rent". Both alternatives are associated with a cost function and switching from rent to buy incurs a constant one-time cost. The goal is to minimize the cost paid compared to the optimal cost, over all possible durations. The problem has been studied for linear and exponential cost functions intensively in the literature.

This paper considers (for the first time) a much more general case, where the cost functions are continuous and only have to satisfy certain mild monotonicity conditions. In the deterministic setting, they give an algorithm that finds the best possible strategy by comparing the cost of the strategy that never buys, the strategy that never rents, and a "middle-ground" strategy which balances buy and rent. In the probabilistic setting, for any given $\varepsilon > 0$, they present an algorithm whose competitive ratio is within $(1+\varepsilon)$-factor from the best possible, if the rent function is unbounded. They show that their proposed probabilistic algorithm runs in time $O(\varepsilon^{-6}\log^4(1/\varepsilon))$.

In the graph (vertex) coloring problem the question is how to color the vertices of a graph such that no two adjacent vertices are of the same color. Trying to minimize the amount of colors used is quite challenging. In the online version, the vertices arrive one by one (together with all the edges adjacent to the already presented vertices) and must be assigned a color immediately and irrevocably. Joanna Chybowska-Sokól et al. [24] work on the online graph coloring problems on the intersection graphs of intervals. Here no two overlapping intervals can be of the same color. If all intervals have length between 1 and σ, the problem is called σ-interval coloring. The authors show that σ-interval coloring is strictly easier than interval coloring and σ-interval coloring is strictly harder than unit interval coloring (where σ is 1).

They present an algorithm that for every $\sigma \in \mathbb{Q}$ with $\sigma \geq 1$ is $1+\sigma$-competitive regarding the asymptotic competitive ratio. Their algorithm constructs small and large blocks (intervals) and then uses counters for these blocks to determine the color for an incoming interval. For $\sigma < 2$, the competitive ratio is less than 3, whereas Kierstead and Trotter [45] showed that there cannot exist a less-than-3-competitive algorithm for (general) interval coloring.

For different values of σ, the authors present corresponding lower bounds. They begin with showing that for every $\sigma > 1$ there is no online algorithm for σ-interval coloring with an asymptotic competitive ratio less than $5/3$. Then they improve the lower bound to $7/4$ for every $\sigma > 2$. Their main result regarding lower bounds is that for every $\varepsilon > 0$ there is $\sigma \geq 1$ such that there is no online algorithm with an asymptotic competitive ratio $5/2 - \varepsilon$. This implies that there is no 2-competitive algorithm for σ-interval coloring (for $\sigma > 2^{78}$). For unit interval graphs, the simple greedy FirstFit algorithm is 2-competitive [30].

A contention resolution scheme abstracts the task in constraint optimization of converting a (random) set-valued solution which is on average feasible for a packing program into one that is always feasible. CRSs are connected to a variety of online and offline problem including online mechanism design, stochastic probing and prophet inequalities.

Most prior work considers a product distribution on the input set of elements (which can arrive offline or online) and studies contention resolution for increasingly general packing constraints. Shaddin Dughmi et al. [29] in ICALP 2020 instead focus on generalizing the input distributions and restrict attention to matroid constraints. Using a set of inequalities, they characterize the class of α-uncontentious distributions. These distributions permit α-competitive offline content resolution for a given matroid. With this characterization they show that content resolution is the natural distributional generalization of base covering. They then establish some basic closure properties for this class.

They also consider online contention resolution on matroids in the random arrival model and tie it to the matroid secretary problem. They show that a γ-competitive secretary algorithm implies that any α-uncontentious distribution permits $\gamma\alpha$-competitive online content resolution. Additionally they reduce the matroid secretary problem to the design of an online contention resolution scheme for a particular uncontentious distribution.

Mark de Berg et al. [26] introduce the online version of the range-assignment problem in ISAAC 2020. In the offline version, there is a set P of n points in $\mathbb{R}^d$, which models devices in a wireless network. By assigning each point $p_i \in P$ a range $r(p_i)$, an algorithm induces a directed communication graph $\mathcal{G}_r$, in which there is a directed edge (p_i, p_j) iff $\text{dist}(p_i, p_j) \leq r(p_i)$. The task is to assign ranges in such a manner, that the graph $\mathcal{G}_r$ has a broadcast tree rooted at the first point p_0, but the cost of the assignment $\sum_{p_i \in P} r(p_i)^\alpha$ (for some constant $\alpha > 1$) is minimized. In this paper, the authors consider for the first time the online version, where the points p_i arrive over time and the range assignment has to be updated at each arrival such that a broadcast tree on the currently inserted points is maintained. At these time steps an algorithm can increase (but not decrease) the ranges of all already seen points.

Their first result is a lower bound for any online algorithm in $\mathbb{R}^1$. They prove that there is a constant $c_\alpha > 1$ such that no online algorithm can be c_α-competitive. For $\alpha = 2$ this results in a lower bound of 1.57. Then the authors focus on two natural strategies and analyze their competitiveness in $\mathbb{R}^1$ and $\mathbb{R}^2$. In $\mathbb{R}^1$ they show that for $\alpha = 2$ the strategy Nearest-Neighbor (nn), which for a point p_i increases the range of its nearest neighbor such that p_i gets covered by its nearest neighbor, is exactly 2-competitive. For $\mathbb{R}^2$ they present a variety of results. First they show that nn is still $O(1)$-competitive for $\alpha = 2$, but the gap between the lower bound of approximately 7.61 and the upper bound of 322 is quite large. Modifying nn, to increase the range to twice the distance between p_i and its nearest neighbor, yields the algorithm 2-Nearest-Neighbor that is 36-competitive. For general $\alpha > 1$ they prove a lower bound of $6\left(1 + (\frac{\sqrt{6}-\sqrt{2}}{2})^\alpha\right)$ for nn. For $\alpha > 2$ they show that nn is better than $F^*_\alpha = \alpha\frac{2^\alpha-3}{2^{\alpha-1}-\alpha}$-competitive. If $\alpha \geq \alpha^* \approx 4.3$, where $\alpha^* = \arg\min F^*_\alpha$, the upper bound improves to approximately 12.94. In fact all their upper bounds also apply to their second strategy Cheapest Increase (ci). This strategy increases the range of the point p_i for which the resulting cost increase to cover the new point is minimal.

The last part of their paper is about the generalization of the problem to general metric spaces, for which they present an $O(\log n)$-competitive algorithm.

References

[1] Susanne Albers and Maximilian Janke. Scheduling in the random-order model. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 68:1–68:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[2] Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private PAC learning implies finite littlestone dimension. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 852–860. ACM, 2019.

[3] Antonios Antoniadis, Carsten Fischer, and Andreas Tönnis. A collection of lower bounds for online matching on the line. In Michael A. Bender, Martin Farach-Colton, and Miguel A. Mosteiro, editors, *LATIN 2018: Theoretical Informatics - 13th Latin American Symposium, Buenos Aires, Argentina, April 16-19, 2018, Proceedings*, volume 10807 of *Lecture Notes in Computer Science*, pages 52–65. Springer, 2018.

[4] C. J. Argue, Anupam Gupta, Guru Guruganesh, and Ziye Tang. Chasing convex bodies with linear competitive ratio. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1519–1524. SIAM, 2020.

[5] Yossi Azar, Ashish Chiplunkar, Shay Kutten, and Noam Touitou. Set cover with delay - clairvoyance is not required. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 8:1–8:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[6] Nikhil Bansal, Haotian Jiang, Sahil Singla, and Makrand Sinha. Online vector balancing and geometric discrepancy. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1139–1152. ACM, 2020.

[7] Nikhil Bansal and Joel H. Spencer. On-line balancing of random inputs. *CoRR*, abs/1903.06898, 2019.

[8] Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 184–193. IEEE Computer Society, 1996.

[9] Yair Bartal, Nova Fandina, and Seeun William Umboh. Online probabilistic metric embedding: A general framework for bypassing inherent bounds. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1538–1557. SIAM, 2020.

[10] Gerdus Benade, Aleksandr M. Kazachkov, Ariel D. Procaccia, and Christos-Alexandros Psomas. How to make envy vanish over time. In Éva Tardos, Edith Elkind, and Rakesh Vohra, editors, *Proceedings of the 2018 ACM Conference on Economics and Computation, Ithaca, NY, USA, June 18-22, 2018*, pages 593–610. ACM, 2018.

[11] Michael A. Bender, Martin Farach-Colton, and William Kuszmaul. Achieving optimal backlog in multiprocessor cup games. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 1148–1157. ACM, 2019.

[12] Aditya Bhaskara, Silvio Lattanzi, Sergei Vassilvitskii, and Morteza Zadimoghaddam. Residual based sampling for online low rank approximation. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 1596–1614. IEEE Computer Society, 2019.

[13] Marcin Bienkowski, Maciej Pacut, and Krzysztof Piecuch. An optimal algorithm for online multiple knapsack. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 13:1–13:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[14] Allan Borodin, Morten N. Nielsen, and Charles Rackoff. (incremental) priority algorithms. *Algorithmica*, 37(4):295–326, 2003.

[15] Prosenjit Bose, Jean Cardinal, John Iacono, Grigorios Koumoutsos, and Stefan Langerman. Competitive online search trees on trees. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1878–1891. SIAM, 2020.

[16] Vladimir Braverman, Petros Drineas, Cameron Musco, Christopher Musco, Jalaj Upadhyay, David P. Woodruff, and Samson Zhou. Near optimal linear algebra in the online and sliding window models. In *61st Annual Symposium on Foundations of Computer Science, FOCS '20*. IEEE Computer Society, 2020.

[17] Sébastien Bubeck, Bo'az Klartag, Yin Tat Lee, Yuanzhi Li, and Mark Sellke. Chasing nested convex bodies nearly optimally. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1496–1508. SIAM, 2020.

[18] Sébastien Bubeck, Yin Tat Lee, Yuanzhi Li, and Mark Sellke. Competitively chasing convex bodies. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019.*, pages 861–868. ACM, 2019.

[19] Sébastien Bubeck and Yuval Rabani. Parametrized metrical task systems. In Jaroslaw Byrka and Raghu Meka, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, August 17-19, 2020, Virtual Conference*, volume 176 of *LIPIcs*, pages 54:1–54:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[20] Mark Bun, Roi Livni, and Shay Moran. An equivalence between private classification and online prediction. In *61st Annual IEEE Symposium on Foundations of Computer Science, FOCS 2020, November 16-19, 2020, Virtual Conference*. IEEE Computer Society, 2020. To appear.

[21] Rodrigo A. Carrasco, Kirk Pruhs, Cliff Stein, and José Verschae. The online set aggregation problem. In Michael A. Bender, Martin Farach-Colton, and Miguel A. Mosteiro, editors, *LATIN 2018: Theoretical Informatics - 13th Latin American Symposium, Buenos Aires, Argentina, April 16-19, 2018, Proceedings*, volume 10807 of *Lecture Notes in Computer Science*, pages 245–259. Springer, 2018.

[22] Jannik Castenow, Björn Feldkord, Till Knollmann, Manuel Malatyali, and Friedhelm Meyer auf der Heide. The online multi-commodity facility location problem. In Christian Scheideler and Michael Spear, editors, *SPAA '20: 32nd ACM Symposium on Parallelism in Algorithms and Architectures, Virtual Event, USA, July 15-17, 2020*, pages 129–139. ACM, 2020.

[23] Moses Charikar and Sudipto Guha. Improved combinatorial algorithms for facility location problems. *SIAM J. Comput.*, 34(4):803–824, 2005.

[24] Joanna Chybowska-Sokól, Grzegorz Gutowski, Konstanty Junosza-Szaniawski, Patryk Mikos, and Adam Polak. Online coloring of short intervals. In Jaroslaw Byrka and Raghu Meka, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, August 17-19, 2020, Virtual Conference*, volume 176 of *LIPIcs*, pages 52:1–52:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[25] Ilan Reuven Cohen, Sungjin Im, and Debmalya Panigrahi. Online two-dimensional load balancing. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 34:1–34:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[26] Mark de Berg, Aleksandar Markovic, and Seeun William Umbo. The online broadcast range-assignment problem. In Yixin Cao, Siu-Wing Cheng, and Minming Li, editors, *31st International Symposium on Algorithms and Computation (ISAAC 2020), December 14–18, 2020, Virtual Conference*, Leibniz International Proceedings in Informatics (LIPIcs), Dagstuhl, Germany, 2020. Schloss Dagstuhl - Leibniz-Zentrum für Informatik. To appear.

[27] Nikhil R. Devanur, Zhiyi Huang, Nitish Korula, Vahab S. Mirrokni, and Qiqi Yan. Whole-page optimization and submodular welfare maximization with online bidders. *ACM Trans. Economics and Comput.*, 4(3):14:1–14:20, 2016.

[28] Nikhil R. Devanur, Kamal Jain, and Robert D. Kleinberg. Randomized primal-dual analysis of RANKING for online bipartite matching. In Sanjeev Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 101–107. SIAM, 2013.

[29] Shaddin Dughmi. The outer limits of contention resolution on matroids and connections to the secretary problem. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 42:1–42:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[30] Leah Epstein and Meital Levy. Online interval coloring and variants. In Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings*, volume 3580 of *Lecture Notes in Computer Science*, pages 602–613. Springer, 2005.

[31] Matthew Fahrbach, Zhiyi Huang, Runzhou Tao, and Morteza Zadimoghaddam. Edge-weighted online bipartite matching. *CoRR*, abs/2005.01929, 2020.

[32] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci.*, 69(3):485–497, 2004.

[33] Dimitris Fotakis, Loukas Kavouras, Grigorios Koumoutsos, Stratis Skoulakis, and Manolis Vardas. The online min-sum set cover problem. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 51:1–51:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[34] Xiangyu Guo, Janardhan Kulkarni, Shi Li, and Jiayi Xian. On the facility location problem in online and dynamic models. In Jaroslaw Byrka and Raghu Meka, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, August 17-19, 2020, Virtual Conference*, volume 176 of *LIPIcs*, pages 42:1–42:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[35] Xiangyu Guo, Janardhan Kulkarni, Shi Li, and Jiayi Xian. On the facility location problem in online and dynamic models. In Jaroslaw Byrka and Raghu Meka, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, August 17-19, 2020, Virtual Conference*, volume 176 of *LIPIcs*, pages 42:1–42:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[36] Anupam Gupta and Roie Levin. The online submodular cover problem. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1525–1537. SIAM, 2020.

[37] Varun Gupta, Ravishankar Krishnaswamy, and Sai Sandeep. Permutation strikes back: The power of recourse in online metric matching. In Jaroslaw Byrka and Raghu Meka, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, August 17-19, 2020, Virtual Conference*, volume 176 of *LIPIcs*, pages 40:1–40:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[38] Zhiyi Huang, Binghui Peng, Zhihao Gavin Tang, Runzhou Tao, Xiaowei Wu, and Yuhao Zhang. Tight competitive ratios of classic matching algorithms in the fully online model. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2875–2886. SIAM, 2019.

[39] Zhiyi Huang, Zhihao Gavin Tang, Xiaowei Wu, and Yuhao Zhang. Online vertex-weighted bipartite matching: Beating 1-1/e with random arrivals. In Ioannis Chatzigiannakis, Christos Kaklamanis,

Dániel Marx, and Donald Sannella, editors, *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018, July 9-13, 2018, Prague, Czech Republic*, volume 107 of *LIPIcs*, pages 79:1–79:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[40] Zhiyi Huang, Zhihao Gavin Tang, Xiaowei Wu, and Yuhao Zhang. Fully online matching II: beating ranking and water-filling. *CoRR*, abs/2005.06311, 2020.

[41] Samin Jamalabadi, Chris Schwiegelshohn, and Uwe Schwiegelshohn. Commitment and slack for online load maximization. In Christian Scheideler and Michael Spear, editors, *SPAA '20: 32nd ACM Symposium on Parallelism in Algorithms and Architectures, Virtual Event, USA, July 15-17, 2020*, pages 339–348. ACM, 2020.

[42] Zhihao Jiang, Debmalya Panigrahi, and Kevin Sun. Online algorithms for weighted paging with predictions. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 69:1–69:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[43] Haim Kaplan, David Naori, and Danny Raz. Competitive analysis with a sample and the secretary problem. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2082–2095. SIAM, 2020.

[44] Thomas Kesselheim and Marco Molinaro. Knapsack secretary with bursty adversary. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPIcs*, pages 72:1–72:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[45] Henry A Kierstead and William T Trotter. An extremal problem in recursive combinatorics. In *Twelfth Southeastern Conference on Combinatorics, Graph Theory and Computing, Vol. II, Baton Rouge, LA, USA, March 1981*, volume 33 of *Congressus Numerantium*, pages 143–153, 1981.

[46] Predrag Krnetić, Darya Melnyk, Yuyi Wang, and Roger Wattenhofer. The k-server problem with delays on the uniform metric space. In Yixin Cao, Siu-Wing Cheng, and Minming Li, editors, *31st International Symposium on Algorithms and Computation (ISAAC 2020), December 14–18, 2020, Virtual Conference*, Leibniz International Proceedings in Informatics (LIPIcs), pages 27:1–27:13, Dagstuhl, Germany, 2020. Schloss Dagstuhl - Leibniz-Zentrum für Informatik. To appear.

[47] Ravi Kumar, Manish Purohit, Zoya Svitkina, and Erik Vee. Interleaved caching with access graphs. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1846–1858. SIAM, 2020.

[48] William Kuszmaul. Achieving optimal backlog in the vanilla multi-processor cup game. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1558–1577. SIAM, 2020.

[49] Silvio Lattanzi, Thomas Lavastida, Benjamin Moseley, and Sergei Vassilvitskii. Online scheduling via learned weights. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1859–1877. SIAM, 2020.

[50] Thodoris Lykouris and Sergei Vassilvitskii. Competitive caching with machine learned advice. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 3302–3311. PMLR, 2018.

[51] Nicole Megow and Lukas Nölke. Online minimum cost matching with recourse on the line. In Jaroslaw Byrka and Raghu Meka, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, August 17-19, 2020, Virtual Conference*, volume 176 of *LIPIcs*, pages 37:1–37:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[52] Aranyak Mehta and Debmalya Panigrahi. Online matching with stochastic rewards. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 728–737. IEEE Computer Society, 2012.

[53] Aranyak Mehta, Bo Waggoner, and Morteza Zadimoghaddam. Online stochastic matching with unequal probabilities. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1388–1404. SIAM, 2015.

[54] Krati Nayyar and Sharath Raghvendra. An input sensitive online algorithm for the metric bipartite matching problem. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 505–515. IEEE Computer Society, 2017.

[55] Boaz Patt-Shamir and Evyatar Yadai. Non-linear ski rental. In Christian Scheideler and Michael Spear, editors, *SPAA '20: 32nd ACM Symposium on Parallelism in Algorithms and Architectures, Virtual Event, USA, July 15-17, 2020*, pages 431–440. ACM, 2020.

[56] Sharath Raghvendra. Optimal analysis of an online algorithm for the bipartite matching problem on a line. *CoRR*, abs/1803.07206, 2018.

[57] Dhruv Rohatgi. Near-optimal bounds for online caching with machine learned advice. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1834–1845. SIAM, 2020.

[58] Mark Sellke. Chasing convex bodies optimally. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 1509–1518. SIAM, 2020.

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

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

[61] Alexander Wei. Better and simpler learning-augmented online caching. In Jaroslaw Byrka and Raghu Meka, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, August 17-19, 2020, Virtual Conference*, volume 176 of *LIPIcs*, pages 60:1–60:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.