SIGACT News Online Algorithms Column 34:
2018 in review

Rob van Stee
University of Siegen
Germany

In this column, I will discuss some papers in online algorithms that appeared in 2018. It is very nice to see many good papers on this topic appearing year after year. I make no claim at complete coverage here, and have instead made a selection.

As always, 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 The k-server problem

The k-server problem has played a major role in the development of the whole field of online algorithms, and remains one of the most well-studied problems in this area. In this problem, requests arrive online in a metric space. The online algorithm has k servers and has to move (at least) one server to the current request before the next one is revealed. The goal is to minimize the total distance traveled by all the servers.

Manasse et al. [40] who introduced this problem in STOC 1988 conjecture that for each metric space, there exists a k-competitive algorithm. Despite decades of work, this conjecture remains open, though it was shown already within a few years by Koutsoupias and Papadimitriou [33] to be true within a factor of 2. Later on, a folklore conjecture arose that for every metric space, a randomized $O(\log k)$-competitive algorithm exists.

Two strong papers on this topic appeared this year. In FOCS 2018, in what is possibly the most impressive result of the year in online algorithms, James Lee [36] presents an $O(\log^6 k)$-competitive randomized algorithm for the k-server problem on any metric space. He shows that a potential-based algorithm for the fractional k-server problem on hierarchically separated trees (HSTs) with competitive ratio $f(k)$ can be used to obtain a randomized algorithm for any metric space with competitive ratio $f^2(k)O(\log^2 k)$. Employing the $O(\log^2 k)$-competitive algorithm for HSTs from Bubeck et al. [15] yields the claimed bound.

This shows for the very first time that for every metric space on which the problem is non-trivial, randomized algorithms give an exponential improvement over deterministic algorithms. Even though this result does not resolve the randomized k-server conjecture entirely, it is a huge improvement compared to what was known.

Nikhil Bansal et al. [11] considered a generalization of the k-server problem in which each server is in its own metric space and each request consists of a point in each metric space. As before, (at least) one server has to travel to a request point each time, and the goal is to minimize the total distance travelled. This paper gives for the first time an $f(k)$-competitive algorithm. That is, the competitive ratio is a function which depends only on k and not on any property of a metric space.

In particular, the authors obtain deterministic and randomized algorithms with competitive ratio $k2^k$ and $O(k^3 \log k)$ respectively. The deterministic bound essentially matches the lower bound of $2^k - 1$ by Koutsoupias and Taylor [34]. The authors also give a $2^{2^{O(k)}}$-competitive deterministic algorithm for weighted uniform metrics, which also essentially matches last year's doubly exponential lower bound for the problem by a subset of the authors [10].

2 Online scheduling

There were several papers on minimizing total weighted flow time this year. This is a problem which in some sense is completely hopeless from a pure competitive ratio perspective, as no algorithm can hope to be competitive. Even on a single machine, there is a lower bound of $\Omega(\sqrt{n})$ where n is the number of job [25]. However, if some parameters of the problem are bounded, positive results can be achieved.

Define P as the maximal ratio between the processing times of any two jobs, and W as the maximal ratio between the weights of any two jobs. Finally, let D be the maximal ratio between the densities of any two jobs, where the density of a job is its weight divided by its size. In STOC 2001, Chekuri et al. [17] had presented an $O(\log^2 P)$-competitive algorithm for a single machine, where P is the ratio between the maximum and the minimum processing time of any job. In SODA 2003, Bansal and Dhamdhere [9] gave an $O(\log W)$-competitive algorithm for this problem.

In this year's FOCS, Yossi Azar and Noam Touitou [5] improve and generalize these results by first presenting an $O(\log P)$-competitive algorithm and a (different) $O(\log D)$-competitive algorithm. (The authors point out that although the parameter D had not previously been considered in the literature, the algorithm of Chekuri et al. can be easily modified to give an $O(\log^2 D)$-competitive algorithm as well.) They then show how to combine these results with the algorithm of Bansal and Dhamdhere to give an algorithm which is $O(\log \min(P, D, W))$-competitive without knowing any of these values in advance. Bansal and Chan [8] had shown in SODA 2009 that no constant-competitive algorithm exists for this problem.

The $O(\log P)$-competitive algorithm works by classifying items geometrically by weight and assigning each class to a separate bin. Within a bin, the jobs are stacked first by weight (highest weight at the top) and then by density (lowest density at the top). At any point in time, the algorithm chooses a bin from which to process the uppermost job. At each point in time, each bin is assigned a score, such that the bin with the highest score is processed.

In Bansal and Dhamdhere [9], the score assigned to each bin is the total weight of the jobs in that bin. In the new algorithm, the score is more complex. All jobs except the top job add their weight to the total score of the bin. The top job adds to the score of the bin either its complete weight, if it has high remaining processing time, or half of it, if it has low remaining processing time. This means that a job can be preempted during processing because its processing time has decreased below the threshold, lowering the score of its bin. This preemption is not due to any external event; no job has been released to trigger it. Similar ideas are used for the $O(\log D)$-competitive algorithm.

Progress was also made on scheduling to minimize weighted flow time on unrelated machines. In ESA 2016, Giorgio Lucarelli et al.[39] had considered a version where the online algorithm can reject some $\varepsilon_r > 0$ fraction (by weight) of the jobs and have machines that are $1 + \varepsilon_s$ as fast as the offline machines, for some $\varepsilon_s > 0$. They showed that this is already enough to achieve a competitive ratio of $O(1/(\varepsilon_s \varepsilon_r)$.

In SPAA 2018, Giorgio Lucarelli et al.[37] (a superset of the previous authors) showed that it is in fact sufficient to reject a 2ε fraction of the total number of jobs to achieve a competitive ratio of $2\left(\frac{1+\varepsilon}{\varepsilon}\right)^2$ for minimizing the *total* flow time. This algorithm sometimes rejects a job other than the one that has just arrived. The authors show that this is necessary, as otherwise there is a lower bound of $\Omega(\Delta)$ even on a single machine. Here Δ is the size ratio (the ratio of largest to smallest job size). (Obviously this lower bound also holds if you cannot reject jobs at all.)

They also consider the speed scaling model, in which machines can be sped up if additional energy is invested, and the goal is to minimize the total weighted flow time plus energy usage. If the power function of machine i is given by $P(s_i(t)) = s_i(t)^\alpha$, where $s_i(t)$ is the current speed of machine i, there is an algorithm which is $O((1 + 1/\varepsilon)^{\alpha/(\alpha-1)})$-competitive that rejects jobs of total weight at most a fraction ε of the total weight of all the jobs. They also give a positive result for jobs with hard deadlines, where the goal is to minimize the total energy usage and no job may be rejected.

In ESA 2018, the same set of authors [38] improved/generalized these results by showing that rejection alone is sufficient for an algorithm to be competitive even for weighted flow time. They presented an $O(1/\varepsilon^3)$-competitive algorithm that rejects at most $O(\varepsilon)$ of the total weight of the jobs. In this algorithm, jobs are assigned (approximately) greedily to machines, and each machine runs the jobs assigned to it using Highest Density First. A job may be rejected if it is running while much heavier jobs arrive or if it is in the queue while very many jobs arrive. The second rule simulates the resource augmentation on the speed.

Also in ESA 2018, Matthias Englert et al. [24] considered the classic online load balancing problem. In this problem, jobs arrive online (not over time) and need to be assigned to machines. The machines are uniform, meaning that they have different speeds. Each job is assigned to one machine and the goal is to minimize the makespan, which is the time it takes until all jobs are processed. Englert et al. consider a setting in which k job can be migrated after the final job has arrived. This setting was previously considered by Albers and Hellwig [1] on parallel (identical) machines. They showed that the competitive ratio of this problem depends on m and grows from 4/3 on two machines to approximately 1.4659 as m tends to infinity.

Englert et al. show that the problem with related machines is harder by presenting a lower bound which is strictly above 1.4659 for large m, as long as $k = o(n)$. They also present an algorithm with competitive ratio between 4/3 and approximately 1.7992, and prove that $k = \Omega(m)$ is necessary to achieve a competitive ratio below 2. The algorithm works by carefully maintaining an imbalance between the machine loads and using a bicriteria approximation algorithm that minimizes the makespan and maximizes the average completion time for certain sets of machines.

Again in ESA 2018, Waldo Gálvez et al. [28] considered machine covering, another problem that I have enjoyed working on in the past. In this problem, the goal is to maximize the minimum load (thus "covering" all the machines with as much load as possible). The authors consider a variant of this problem with migration as above, but in this paper, migration is allowed after every single job arrival. That is, we are allowed to reassign some jobs as long as their total size is (at most) proportional to the processing time of the arriving job. The proportionality constant is called the migration factor of the algorithm. The authors present a $(4/3+\varepsilon)$-competitive algorithm with migration factor $\tilde{O}(1/\varepsilon^3)$. It runs an adaptation of LPT at every job arrival. Since the new job can cause a complete change of the assignment of smaller jobs, a low migration factor is achieved by carefully exploiting the highly symmetric structure obtained by the rounding procedure.

In STOC 2018, Sungjin Im et al. [31] considered load balancing on related machines. They give a constant competitive algorithm for optimizing any ℓ_p-norm for scheduling on related machines. The only previously known result was for the makespan norm. Additionally, they consider vector scheduling. In vector scheduling, vectors need to be assigned to machines and the goal is to minimize the makespan, which is defined as the maximum load across all dimensions and machines. Im et al. show that there is a sharp contrast between the case where the speed of a machine depends on the dimension and the case where it does not. In fact they show that the first case is equivalent to scheduling on unrelated machines and the second case is equivalent to scheduling identical machines. The results are also extended to ℓ_p norms of the machine loads.

Finally, in ESA 2018, Alon Eden et al. [23] considered truthful prompt scheduling. They consider an algorithmic game theory setting and give an online mechanism for minimizing the sum of weighted completion times. The mechanism is prompt, meaning that the mechanism immediately decides when an incoming job will be processed, without using preemption. It does not mean that incoming jobs will be served as soon as possible (if they are served), as it is impossible to be competitive in that setting.

In the mechanism presented here, each job (agent) is presented with a menu of options when it arrives, which specify an interval for the job and a cost to be run in that interval. These menus are anonymous and do not depend on the agent that arrives; in particular, they do not depend on the size of the current job (which the agent could try to misrepresent in order to be served sooner). This is a very nice idea and the authors prove that this mechanism achieves logarithmic competitive ratios in several settings of the parameters. In addition, it is shown that these ratios are (essentially) optimal.

3 Online matching problems

There are several models for online matching. In the most commonly studied version, one set L of vertices is given in advance (the offline vertices), and the other set R arrives online. Karp et al. [32] introduced this problem in STOC 1990. In SODA 2018, Ilan Reuven Cohen et al. [18] considered randomized matching in regular graphs. They showed that for this problem, a competitive ratio of $1-O(\sqrt{\log d}/\sqrt{d})$ can be achieved in expectation on d-regular graphs, and a ratio of $1-O(\log(n)/\sqrt{d})$ can be achieved with high probability, as well as guaranteeing each vertex a probability of being matched tending to 1. These results are complemented by a randomized lower bound of $1-O(1/\sqrt{d})$ using Yao's principle. The algorithm works by *marking* a superset of the matched vertices to ensure that each offline vertex has a fixed probability of exactly $1/d$ of being marked.

Also in SODA, Bernstein et al. [12] considered this problem with amortized replacements. The goal is to maintain a maximum matching while minimizing the number of changes (replacements) to the matching. They show that a greedy algorithm that always takes the shortest augmenting path (SAP) from the newly inserted vertex uses at most $O(\log^2 n)$ replacements per insertion, where n is the total number of vertices inserted. This improves on the previously best result of $O(\sqrt{n})$ from Bosek et al. [13] in FOCS 2014 and almost matches the lower bound of $\Omega(\log n)$. The result also works for the capacitated assignment problem, where each offline vertex has a capacity to serve a number of online vertices.

In STOC, Zhiyi Huang et al. [29] introduce a fully online model in which all vertices online. On the arrival of a vertex, its incident edges to previously-arrived vertices are revealed. Each vertex has a deadline that is after all its neighbors' arrivals. If a vertex remains unmatched until its deadline, the algorithm must then irrevocably either match it to an unmatched neighbor, or leave it unmatched. The model generalizes the existing one-sided online model and is motivated by applications including ride-sharing platforms, real-estate agency, etc.

They show that the Ranking algorithm by Karp et al. [32] is 0.5211-competitive for this model on general graphs, getting a bound above 0.5 for the first time. For bipartite graphs, the competitive ratio of Ranking is shown to be between 0.5541 and 0.5671. Additionally, a lower bound of $0.6317 < 1 - 1/e$ for bipartite graphs shows that this fully online model is strictly harder than the previous model.

A subset of the authors considered the standard (one-sided online) version of this problem in ICALP [30]. They introduce a weighted version of Karp's algorithm and prove a competitive ratio of 0.6534 for vertex-weighted online bipartite matching when the online vertices arrive in random (rather than adversarial) order. The algorithm uses a gain sharing function which depends on two variables. Essentially, offline vertices offer a larger portion of their weights to the online vertices as time goes by, and each online vertex matches the neighbor with the highest offer at its arrival.

In SoCG, Sharath Raghvendra [42] considered the famous problem of online matching on the line (see also last year's column [45]). He shows that the algorithm from his joint paper with Nayyar [41] from FOCS 2017 is in fact $O(\log n)$-competitive, thus proving it is at least as good as the work function algorithm. Of course, whether a constant competitive algorithm exists remains an open question!

4 Bin and vector packing

Vector scheduling and vector bin packing was considered by Yossi Azar et al. [4] in SODA. In vector bin packing, vectors with entries between 0 and 1 need to be assigned to bins of size B and the goal is to minimize the number of bins used. Previous work of an overlapping set of authors [3] had given an upper bound of $\tilde{O}(d^{1/(B-1)})$ and a lower bound of $\Omega(d^{1/B-\varepsilon})$ for any $\varepsilon > 0$. The new paper closes the gap (up to log factors) by giving an algorithm with competitive ratio $\tilde{O}(d^{1/B})$. For $B = 2$, the result is improved from $O(d)$ to $\tilde{O}(\sqrt{d})$.

They also give a randomized lower bound for vector scheduling which matches the best known (and deterministic) upper bound for this problem, which is $\Theta(\log d/\log\log d)$.

János Balogh et al. [6] considered online bin packing in ESA. They give a 1.578-competitive online algorithm for this classic problem by setting up different linear programs for every relevant case. In a few years, the gap for this problem has now been reduced from about 0.05 to about 0.038. Will we ever know the true ratio of this problem?

5 Online facility location

Marek Cygan et al. [19] considered a dynamic version of the online facility location problem, where clients as well as facilities may depart. They give an optimal $O(\log a/\log\log a)$-competitive algorithm, where a is the number of active clients at the end of the input sequence. They also consider the uniform capacitated version, where each facility has a capacity c. If deletions are not allowed, they give an optimal $O(\log n/\log\log n)$-competitive algorithm where n is the length of the sequence. For the more challenging case with deletions, they present an $O(\log m + \log c \log n)$-competitive algorithm, where m is the number of points in the input metric.

Bjrn Feldkord and Friedhelm Meyer auf der Heide [26] consider another variant of the facility location problem, in which the online algorithm is allowed to adapt the position of the facilities for costs proportional to the distance by which the position is changed. They give algorithms for Euclidean space of arbitrary dimension and show that these are asymptotically optimal for the line. The competitive ratios of the algorithms do not depend on the number of clients.

In the streaming model, the order of the stream can significantly affect the difficulty of a problem. Harry Lang [35] considered a t-semirandom stream, which was introduced as an interpolation between random-order ($t = 1$) and adversarial-order ($t = n$) streams where an adversary intercepts a random-order stream and can delay up to t elements at a time. IITK Sublinear Open Problem 15 asks to find algorithms whose performance degrades smoothly as t increases. Lang shows that the known online facility location algorithm achieves an expected competitive ratio of $O(\log t/\log\log t)$ and a matching lower bound. He also gives an application to streaming for k-median clustering.

6 Various problems

Nikhil Bansal et al. [7] considered the convex body chasing problem in SODA 2018. In this problem, we are given an initial point $v_0 \in \mathbb{R}^d$ and an online sequence of n convex bodies $F_1, \ldots, F_n$. When we receive F_i, we are required to move inside F_i. Our goal is to minimize the total distance traveled. This fundamental online problem was first studied by Friedman and Linial [27]. They proved an $\Omega(\sqrt{d})$ lower bound on the competitive ratio, and conjectured that a competitive ratio depending only on d is possible. However, despite much interest in the problem, the conjecture remained wide open (but see [14] which arrived just before press time).

Bansal et al. consider the setting in which the convex bodies are nested: $F_1 \supset \ldots \supset F_n$. The motivation for this is that understanding this problem is a necessary step towards the larger goal of extending the online LP framework of Buchbinder and Naor [16] more broadly beyond packing and covering LPs. For example, it is unclear how to do this even for seemingly simple formulations such as k-server on depth-2 HSTs or Metrical Task Systems on a line. The nested convex body chasing problem corresponds to solving online LPs with arbitrary constraints (with both positive and negative entries) and a specific type of objective.

Moreover, this setting retains much of the difficulty of the general setting and captures an essential obstacle in resolving Friedman and Linials conjecture. The authors give a $f(d)$-competitive algorithm for chasing nested convex bodies in $\mathbb{R}^d$.

Nikhil Devanur et al. [21] considered online auctions. They showed that it is not possible to give any deterministic individually rational mechanism for this problem which has a finite competitive ratio for the social welfare if the auctioneer has to decide immediately after each item arrives how to allocate it. This holds even for the most restricted nontrivial version of the problem in which there are only two item types and two unit-demand bidders, and even if the payments are allowed to be computed after knowing how many items arrived.

References

[1] Susanne Albers and Matthias Hellwig. On the value of job migration in online makespan minimization. In Leah Epstein and Paolo Ferragina, editors, *Algorithms - ESA 2012 - 20th Annual European Symposium, Ljubljana, Slovenia, September 10-12, 2012. Proceedings*, volume 7501 of *Lecture Notes in Computer Science*, pages 84–95. Springer, 2012.

[2] Yossi Azar, Hannah Bast, and Grzegorz Herman, editors. *26th Annual European Symposium on Algorithms, ESA 2018, August 20-22, 2018, Helsinki, Finland*, volume 112 of *LIPIcs*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

[3] Yossi Azar, Ilan Reuven Cohen, Seny Kamara, and F. Bruce Shepherd. Tight bounds for online vector bin packing. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 961–970. ACM, 2013.

[4] Yossi Azar, Ilan Reuven Cohen, and Debmalya Panigrahi. Randomized algorithms for online vector load balancing. In Czumaj [20], pages 980–991.

[5] Yossi Azar and Noam Touitou. Improved online algorithm for weighted flow time. *CoRR*, abs/1712.10273, 2017.

[6] János Balogh, József Békési, György Dósa, Leah Epstein, and Asaf Levin. A new and improved algorithm for online bin packing. In Azar et al. [2], pages 5:1–5:14.

[7] Nikhil Bansal, Martin Böhm, Marek Eliáš, Grigorios Koumoutsos, and Seeun William Umboh. Nested convex bodies are chaseable. In Czumaj [20], pages 1253–1260.

[8] Nikhil Bansal and Ho-Leung Chan. Weighted flow time does not admit o(1)-competitive algorithms. In Claire Mathieu, editor, *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, NY, USA, January 4-6, 2009*, pages 1238–1244. SIAM, 2009.

[9] Nikhil Bansal and Kedar Dhamdhere. Minimizing weighted flow time. *ACM Trans. Algorithms*, 3(4):39, 2007.

[10] Nikhil Bansal, Marek Eliáš, and Grigorios Koumoutsos. Weighted k-server bounds via combinatorial dichotomies. In Umans [44], pages 493–504.

[11] Nikhil Bansal, Marek Eliáš, Grigorios Koumoutsos, and Jesper Nederlof. Competitive algorithms for generalized k-server in uniform metrics. In Czumaj [20], pages 992–1001.

[12] Aaron Bernstein, Jacob Holm, and Eva Rotenberg. Online bipartite matching with amortized replacements. In Czumaj [20], pages 947–959.

[13] Bartlomiej Bosek, Dariusz Leniowski, Piotr Sankowski, and Anna Zych. Online bipartite matching in offline time. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 384–393. IEEE Computer Society, 2014.

[14] S. Bubeck, Y. Tat Lee, Y. Li, and M. Sellke. Competitively Chasing Convex Bodies. *ArXiv e-prints*, November 2018.

[15] Sébastien Bubeck, Michael B. Cohen, James R. Lee, Yin Tat Lee, and Aleksander Madry. k-server via multiscale entropic regularization. *CoRR*, abs/1711.01085, 2017.

[16] Niv Buchbinder and Joseph Naor. Online primal-dual algorithms for covering and packing. *Math. Oper. Res.*, 34(2):270–286, 2009.

[17] Chandra Chekuri, Sanjeev Khanna, and An Zhu. Algorithms for minimizing weighted flow time. In Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis, editors, *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 84–93. ACM, 2001.

[18] Ilan Reuven Cohen and David Wajc. Randomized online matching in regular graphs. In Czumaj [20], pages 960–979.

[19] Marek Cygan, Artur Czumaj, Marcin Mucha, and Piotr Sankowski. Online facility location with deletions. In Azar et al. [2], pages 21:1–21:15.

[20] Artur Czumaj, editor. *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*. SIAM, 2018.

[21] Nikhil R. Devanur, Balasubramanian Sivan, and Vasilis Syrgkanis. Truthful multi-parameter auctions with online supply: an impossible combination. In Czumaj [20], pages 753–769.

[22] Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors. *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*. ACM, 2018.

[23] Alon Eden, Michal Feldman, Amos Fiat, and Tzahi Taub. Truthful prompt scheduling for minimizing sum of completion times. In Azar et al. [2], pages 27:1–27:14.

[24] Matthias Englert, David Mezlaf, and Matthias Westermann. Online makespan scheduling with job migration on uniform machines. In Azar et al. [2], pages 26:1–26:14.

[25] Leah Epstein and Rob van Stee. Lower bounds for on-line single-machine scheduling. *Theoretical Computer Science*, 299(1-3):439–450, 2003.

[26] Björn Feldkord and Friedhelm Meyer auf der Heide. Online facility location with mobile facilities. In Scheideler and Fineman [43], pages 373–381.

[27] Joel Friedman and Nathan Linial. On convex body chasing. *Discrete & Computational Geometry*, 9:293–321, 1993.

[28] Waldo Gálvez, José A. Soto, and José Verschae. Symmetry exploitation for online machine covering with bounded migration. In Azar et al. [2], pages 32:1–32:14.

[29] Zhiyi Huang, Ning Kang, Zhihao Gavin Tang, Xiaowei Wu, Yuhao Zhang, and Xue Zhu. How to match when all vertices arrive online. In Diakonikolas et al. [22], pages 17–29.

[30] 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 fuer Informatik, 2018.

[31] Sungjin Im, Nathaniel Kell, Debmalya Panigrahi, and Maryam Shadloo. Online load balancing on related machines. In Diakonikolas et al. [22], pages 30–43.

[32] Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In Harriet Ortiz, editor, *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 352–358. ACM, 1990.

[33] Elias Koutsoupias and Christos H. Papadimitriou. On the k-server conjecture. *J. ACM*, 42(5):971–983, 1995.

[34] Elias Koutsoupias and David Scot Taylor. The CNN problem and other k-server variants. *Theor. Comput. Sci.*, 324(2-3):347–359, 2004.

[35] Harry Lang. Online facility location against a t-bounded adversary. In Czumaj [20], pages 1002–1014.

[36] James R. Lee. Fusible hsts and the randomized k-server conjecture. *CoRR*, abs/1711.01789, 2017.

[37] Giorgio Lucarelli, Benjamin Moseley, Nguyen Kim Thang, Abhinav Srivastav, and Denis Trystram. Online non-preemptive scheduling on unrelated machines with rejections. In Scheideler and Fineman [43], pages 291–300.

[38] Giorgio Lucarelli, Benjamin Moseley, Nguyen Kim Thang, Abhinav Srivastav, and Denis Trystram. Online non-preemptive scheduling to minimize weighted flow-time on unrelated machines. In Azar et al. [2], pages 59:1–59:12.

[39] Giorgio Lucarelli, Nguyen Kim Thang, Abhinav Srivastav, and Denis Trystram. Online non-preemptive scheduling in a resource augmentation model based on duality. In Piotr Sankowski and Christos D. Zaroliagis, editors, *24th Annual European Symposium on Algorithms, ESA 2016, August 22-24, 2016, Aarhus, Denmark*, volume 57 of *LIPIcs*, pages 63:1–63:17. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

[40] Mark S. Manasse, Lyle A. McGeoch, and Daniel Dominic Sleator. Competitive algorithms for server problems. *J. Algorithms*, 11(2):208–230, 1990.

[41] Krati Nayyar and Sharath Raghvendra. An input sensitive online algorithm for the metric bipartite matching problem. In Umans [44], pages 505–515.

[42] Sharath Raghvendra. Optimal analysis of an online algorithm for the bipartite matching problem on a line. In Bettina Speckmann and Csaba D. Tóth, editors, *34th International Symposium on Computational Geometry, SoCG 2018, June 11-14, 2018, Budapest, Hungary*, volume 99 of *LIPIcs*, pages 67:1–67:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

[43] Christian Scheideler and Jeremy T. Fineman, editors. *Proceedings of the 30th on Symposium on Parallelism in Algorithms and Architectures, SPAA 2018, Vienna, Austria, July 16-18, 2018*. ACM, 2018.

[44] Chris Umans, editor. *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*. IEEE Computer Society, 2017.

[45] Rob van Stee. SIGACT news online algorithms column 32: 2017 in review. *SIGACT News*, 48(4):100–109, 2017.