Appendix A: Dynamic programming for sale of ad space

Our simple dynamic programming is explained in Algorithm 3. We assume (without loss) that the m ads are indexed so that all ads from the same advertiser lie in a contiguous range. That is, no ad of one advertiser comes between two ads from another advertiser. Then the subproblem for our dynamic program is $\texttt{sub}(i, h', k')$, which gives the optimal welfare while using ads from index i to m, allocating at most h' ads from distinct advertisers and using at most k' lines. The solution to the full allocation problem is then $\texttt{sub}(1, h, k)$. We use $l(\text{ad}_i)$ as a notation for the number of lines for ad_i and $W(\text{ad}_i)$ for the expected (declared) value of ad_i to its associated advertiser.

Algorithm 3: (Dynamic program for winner determination)

Input: ads $\{\text{ad}_i\}_{i\in m}$, index I, number of ads to be allocated h' and maximum number of available lines k'

initialize bestWF $\leftarrow 0$.
if $I > m$ *or* $h' < 1$ *or* $k' < 1$ **then**
  return 0
else
  for $i = I$ ***to*** m **do**
    Let j be next ad from different advertiser than i.
    if $bestWF < W(ad_i) + sub(j, h' - 1, k' - l(ad_i))$ **then**
      bestWF $\leftarrow W(\text{ad}_i) + \text{sub}(j, h' - 1, k' - l(\text{ad}_i))$
    Let $\tilde{\pi}_i = \pi_i + \Delta$ for $i \in S$, $\tilde{\pi}_i = \pi_i$ for $i \in N \setminus S$, and $\tilde{\pi}_0 = w(N) - \sum_{i\in N} \tilde{\pi}_i$.
  return bestWF

Appendix B: Benchmark core pricing algorithms and other auctions

Here is a detailed list of payment rules/auctions we implemented in this paper:

i. The Vickrey-Clarke-Groves auction (`VCG`): See Vickrey 1961;Clarke 1971;Groves 1973.

ii. The Generalized Second Price auction with optimal welfare allocation (`GSP with Optimal`): this auction uses the optimal welfare allocation. For payments, similar to traditional GSP, it prices each ad according to the p_{click} times bid of the subsequent ad (the last ad will be priced by the best ad that was not assigned and can be fit within the line count limit).

iii. The Generalized Second Price auction with greedy allocation (`GSP with Greedy`): for the allocation, it greedily allocates ads based on p_{click} times bids. For payments, similar to GSP, uses the next best ad for pricing. Greedy GSP has faster runtime, since it does need not to call the winner determination oracle, but has worse revenue performance relative to Optimal GSP (as we will see in our experimental results).

iv. Minimum revenue core payment rule (`Min Rev Core`; Day and Raghavan 2007): this is the first and simplest heuristic algorithm that finds a minimum revenue core point, given access to an oracle for the winner determination problem (with truncated values). This algorithm is based on a heuristic called Core Constraint Generation (CCG). The simple version in Day and Raghavan 2007 starts from an initial small LP for minimizing revenue (which is basically the hyper-cube when payments are above VCG prices and below bids), and in each iteration finds the most violated core constraint by the current point (by sending a query to the winner determination oracle of Definition 6) and adds this constraint to the LP. It then re-solves the LP to find the next point, and iterates until it finds a feasible core point. We use Matlab's large-scale LP solver based on the interior-point methods (Zhang 1998) for the LP-solving part of this algorithm.

v. Quadratic core payment rule (`Quad Core`; Day and Cramton 2012): this algorithm finds the closest minimum revenue core point to VCG prices. It is again a heuristic algorithm that first uses CCG and finds the minimum revenue core point at each iteration similar to Day and Raghavan 2007. Then, by fixing this revenue, it searches for another point in the current feasible polytope of core candidates (with one additional constraint for fixing the revenue) that has minimum ℓ_2-distance to VCG. This search is done using convex quadratic programming. The algorithm iterates over this procedure until it finds a feasible core point. We use Matlab's large-scale LP solver based on the interior-point methods (Zhang 1998) for the linear programming part, and Matlab's large scale interior-point method for solving convex quadratic programming (Mehrotra 1992, Gould and Toint 2004) for the quadratic programming part of each iteration.

vi. Vaidya's cutting plane method (`Vaidya Min Rev`; Vaidya 1989, 1996): one approach to find a minimum revenue core point in polynomial-time is to solve the LP for the minimum revenue directly, as we have access to a separation oracle for the core polytope. As a reminder, this oracle is essentially the allocation algorithm for the winner determination problem with truncated values, as in Definition 6. Vaidya's volumetric cutting plane method is a fast algorithm that makes efficient use of the separation oracle. If n is the number of bidders, it has oracle complexity $O(n\log(n))$ and computational complexity $O(n^4)$. For details on different steps of this algorithm and its analysis see Bubeck et al. 2015. We implemented Vaidya's algorithm by following the steps in Section 2.3 of Bubeck et al. 2015 to be used in our numerical experiments.

vii. Our bidder optimal core pricing algorithm (`Fast Core`): we implemented our core pricing rule following the steps of Algorithm 1, which uses Algorithm 2 as a subroutine.

REMARK 5. Note that we studied above auctions *without* reserve prices. Importantly, tuned reserve prices are commonly used to boost revenue. However, as reserves can be applied to all seven of these auctions, we choose not to include them in our experiment in order to focus on the impact of the auction pricing rule.

REMARK 6. In fact, the computational complexity of Vaidya can even be improved further, and the recent breakthrough by Lee et al. 2015 shows that it can essentially (up to logarithmic factors) be brought down to $O(n^3)$. As we will elaborate in Section C.4, we already faced several practical complications and obstacles to implement Vaidya for our application (mostly due to the sensitivity of its performance to various parameters of the algorithm to make use of volumetric barrier). Hence, we left implementing the algorithm of Lee et al. 2015 and adapting it for our application as a future direction.

Appendix C: Practical considerations and limitations in our experimental study

Here we note a few important practical notes that apply to our numerical results of this section, together with potential interpretations or road-maps on how to deal with them based on various work in the literature. Further and deeper discussion on these points is beyond the scope of this work and we leave them as interesting open directions for future work.

C.1. Bid collection vs. true valuations

During bid collection, the sponsored search platform was running its own native auction (which roughly speaking is a variant of the generalized second price auction with optimized reserve prices). These bids by no means are guaranteed to be truthful bids, neither are collected through a controlled experiments in which bidders are aware of changing the auction to a core selecting auction or any other auction that we are simulating in this section. Moreover, in online advertising, it is difficult to evaluate an advertiser's true valuation of a click from the submitted bids, because of several fundamental reasons: (1) Importantly, advertisers usually run sophisticated learning algorithms to bid based on their past experiences with the platform, which leads to complicated bid shading mechanics, (2) Indeed, the advertiser itself might not know the true valuations for the clicks received from the search engine, as the quality of clicks differs and is dependent on the user, publisher, and other contexts, (3) Advertisers might not be utility maximizers, for example they might be maximizing clicks given a budget, or might be valuing the conversion (when a defined transaction such as a sale or subscribing happens from the user after the click) or just a visit to their page, and hence their behavior diverge from the classic quasi-linear rational models in microeconomics, and finally (4) Even if the appropriate metrics were known, the platform typically does not get to observe them accurately. Understanding and modeling true advertisers' behavior is still an important open problem for ad auction research and industry, and is beyond the scope of this paper (Edelman and Ostrovsky 2007, Xu et al. 2013, Sayedi 2018).

Nevertheless, in the absence of knowing the exact behavior of advertisers and the possibility of running a controlled exclusive experiment, the bidding numbers are our best estimates for the advertisers' true values for clicks. Therefore, when using bidding data to evaluate our algorithms, we implicitly assumed truthful bidding and took each advertiser's bid as a proxy for their value. We made this

choice in part due to necessity, and in part because it is a common practice in search advertising auction industry. As a minor note, there are also strong empirical evidences for modeling advertisers as ROI (Return of Investment) constrained value maximizers (e.g., see Aggarwal et al. 2006, Wilkens et al. 2017), under which GSP is provably a truthful auction (Wilkens et al. 2017). Yet, these results heavily rely on myopic rational behavior for advertisers, which might not be the case in practice.

C.2. Non-truthfulness in core auctions and short/long-term incentive issues

One important property that core selecting auctions lack is truthfulness. So, even though in our numerical experiments the reported bids are acceptable proxies for the true valuations, it is critical to understand the behavior of bidders, both in short-term and long-term, and how they respond to the non-truthfulness of the auction. This response can have implications on the revenue of the auction. For example, when bidders bid strategically or run a learning algorithm to respond, the platform's revenue could be hurt. We propose the following interpretations and methodologies to study this phenomena. Digging deeper in some of these methodologies is beyond the scope of this work and we leave as future research directions:

- Running a bidder optimal core selecting auction imposes a natural full information Nash equilibrium, where bidders truncate their values by the utility they get from the bidder optimal core pricing based on the the core with respect to the *true* valuations (Day and Milgrom 2008). Moreover, the revenue of the seller at this equilibrium is equal the revenue of the bidder optimal core pricing with respect to the true valuations. As we are running bidder optimal core selecting auctions and advertises have not responded to this auction (so, bids are proxies of true valuations), one can therefore interpret our experimental results as evaluating the *short-term* impact of employing core pricing, modulo the assumption that advertisers are playing the aforementioned full information Nash equilibrium above.

- Assuming that advertisers play the full information Nash equilibrium can be problematic in sponsored search auctions, as these markets are highly uncertain and information about competitors is incomplete. In such an environment, an advertiser might run a learning algorithm or some other dynamic responding mechanism to bid. However, there are evidences that bidder optimal core auctions "guide the advertisers" who run natural dynamics, in a way that advertisers converge fast to this full information Nash equilibrium, without actually needing complete information. To see this, consider the following two-step process: in the first step, advertisers bid truthfully (as they have no information about other competitors) and each winner obtains some utility. Now, as the auction is a bidder optimal core auction, these utilities will reveal the full information Nash equilibrium mentioned earlier. In fact, each bidder only needs to play a truncated strategy that shades the bid of each package by the utility obtained in the first step (which is a somehow natural bid shading algorithm), and in this way the full information Nash equilibrium will be played in the second step. One can also think

of other generalizations of the mentioned two-step dynamic, where the bidders only shade their bid by truncating based on a fraction of the utility of the previous round. We conjecture this simple dynamics, as well as other dynamics such as iterative best response, converge to the full information Nash equilibrium in the sale of ad space problem, and we leave investigating further as an open question (interestingly, there are examples showing that for a general combinatorial auction this dynamics does not converge, and at the same time there is a simple proof showing that for simple settings such as single-minded combinatorial auctions this dynamics converges to the full information Nash equilibrium).

- Another possible approach to interpret the the long-term effects of the incentive issues of core auctions is to focus on an incomplete information equilibrium concept such as Bayes Nash Equilibrium (BNE). While characterizing closed-form equilibria is intractable (and probably impossible), one can use the recent progress on computational methods to approximate the BNE of core selecting auctions (Bosshard et al. 2017, Lubin and Parkes 2009, Lubin et al. 2015, Bünz et al. 2015, 2018b,a), in order to obtain an educated guess ball-park characterization of how far we might expect the revenue to be from the simulation results in practice (basically after platform uses this auction and advertisers respond to it in a way that they approximately land in the aforementioned BNE). Bünz et al. 2018a,b used the computational search approach to help with better understanding the quadratic core payment rule, as well as designing an automatic search for "better" core payment rules. This paper cannot be used in a blackbox fashion to assess our core pricing rule (as one needs to run the computational search approach for our core pricing rule. Also they have considered different domains than ours and revenue is very domain dependent). Yet, it can be used to obtain very rough estimates of how much revenue reduction we might face at the (approximate) BNE. By looking at the revenue column of Tables 1-7 in Bünz et al. 2018a,b, it seems the revenue ratio of quadratic payment rule over VCG is a number between 0.88-1.44 among 7 different domains (which is averaged at 11.27% improvement). Our numerical experiments suggest an improvement around 15% (see Table 1). One might try to repeat the same approach, but tailored to our core selection algorithm and our domain, and refine the 26.5% improvement over VCG that our experimental results suggest for the revenue of our core pricing algorithm. We leave this refined experimental study as an interesting future direction.

C.3. Using approximate winner determination for faster auctions

We described a dynamic program for generating a welfare-optimal slate of ads, and showed that it was feasible within the time constraints imposed by a production advertising platform for a realistic number of ads and lines. However, if the number of advertisers or lines increases, and/or there is a change on the assumptions that can be made on buyer utilities, it may turn out that the dynamic programming solution is not always feasible in practice. If not, we note that one can replace the optimal allocation with an approximate Maximal-In-Range allocation (see Dobzinski and Nisan 2007)

and get the same result as in this paper, but for the core polytope restricted to that range. In other words, if one can compute the optimal allocation within a restricted range of allocations (e.g., those that allocate at most 4 ads, or that only show ads of the same size, etc.), then our algorithm – using such a restricted welfare maximization oracle – will generate core payments for the auction that restricts outcomes to lie in that range.

C.4. Comments on implementing cutting plane methods

We observed the following issues while implementing Vaidya's algorithm (Vaidya 1989, 1996) for the sale of ad space application (we use notations used in Section 2.3 of Bubeck et al. 2015), which we believe makes this algorithm (and other cutting plane methods with similar operations) impractical for our purpose:

i. Vaidya's algorithm hugely depends on the precision handling, where precision controls "how much the final point is close to a minimum revenue core point". In fact, it is very susceptible to overflow as it needs to compute the logarithm function for different "slack" terms, and overflow easily happens when these slack terms approach to zero. Hence, in practice, the precision has to be very lax in order for the algorithm to converge.

ii. Vaidya's algorithm convergence hugely depends on different parametric choices (e.g., when to break the algorithm based on the volume of the search region, choice of parameter β in each iteration, etc.). So, even with a lax precision it takes a long time to converge in some cases.

In implementing Vaidya's algorithm, we note that as we increase the precision of the solution slightly, its running time degrades significantly. Our experiments show that the running time scales by a factor comparable to $O(\frac{1}{\epsilon})$ in order to obtain an small precision of ϵ. This is in contrast to our algorithm where running time only scales by $O(\log(\frac{1}{\epsilon}))$ to get the precision ϵ (and our experiments validate this fact as well). Another important factor in practice is the ignored constants of the big O notation in the running time. We again argue that comparing to our algorithm Vaidya's algorithm has much larger constants in its running time, as suggested by our experiments and carefully implementing the algorithm following the recipe in Section 2.3 of Bubeck et al. 2015). As a result, even obtaining a point with a rather large precision of $\epsilon = 0.1$ requires a long processing time.

To better understand the reasons behind the precision sensitivity, which are also true for other barrier-based cutting plane methods such as Lee et al. 2015, consider the following. The Vaidya's algorithm has three main parameters that govern the precision of its final solution (and therefore its running time). The first parameter is the stopping threshold for the volume of the search polytope (measured indirectly using the volumetric barrier), below which we stop reducing the feasible region and terminate. The second parameter governs how small the leverage score of a hyperplane associated to a face of the search polytope should be to be removed (see Section 2.3.2 of Bubeck et al. 2015, line (1) in the description of the algorithm for more details). Finally the last parameter is the standard

floating point precision which is used for floating point operations in practice. Our experiments (and further playing around with Vaidya's algorithm to optimize its running time) suggest that Vaidya's running time is very sensitive to the choice of the first and second parameters above, mostly due to the logarithm function used in the volumetric barrier.

Appendix D: Future directions and open problems

Here is a list of a few concrete theoretical and practical future directions:

i. Our fast core pricing algorithm (Algorithm 1) does not target any *global* objective function, which is in contrast to other traditional methods such as Day and Raghavan (2007) (minimizing ℓ_1 distance from VCG) and Day and Cramton (2012) (minimizing ℓ_2 distance from VCG). Can one use the way we explore the core polytope efficiently to optimize a tractable (maybe convex) objective function in a fast fashion?

ii. Give the practical considerations in Section C.4, how can one adapt the state of the art cutting plane methods such as Lee et al. (2015) to work for the sale of ad space problem? Our simulations suggest these algorithms are very sensitive to different precision parameters. Designing a robust version of them sounds like a roadmap to approach this challenge.

iii. In sponsored search, advertisers either use sophisticated online learning algorithms, or are relying on automated bidding softwares provided by the platform (Aggarwal et al. 2019). What can we say about the way they respond to a core selecting auction, in comparison to GSP auction and other auction formats that are common in sponsored search?

iv. Designing auctions for video ads is another combinatorial setting that can potentially benefit from a combinatorial auction such as core selecting. An interesting open problem is to use the techniques in our paper to tackle this multi-billion dollar industry problem.

Appendix E: More experimental results for various line counts

We report the revenue, fairness, running time and query complexity of different algorithms in our experiments for line counts 25, 30, 35, and 45. For revenue results, see Figures 7, 8, 9, and 10. For running time results, see Figures 11, 12, 13, and 14. For query complexity results, see Figures 15, 16, 17, and 18. Finally, for fairness results, see Figures 19, 20, 21, and 22.

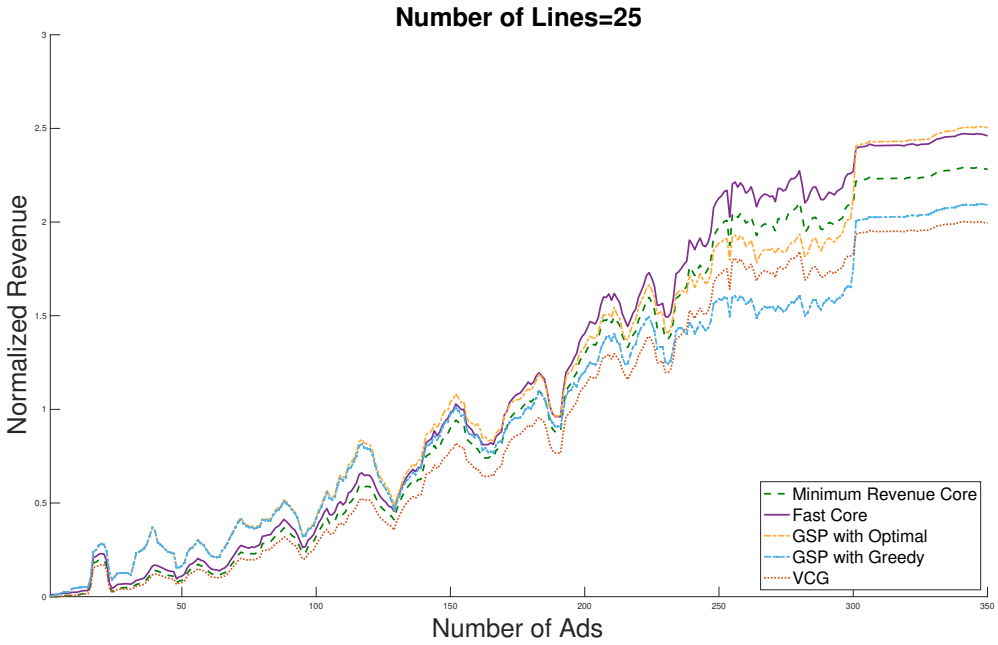

Figure 7 Normalized revenues versus total number of ads for line count=25.

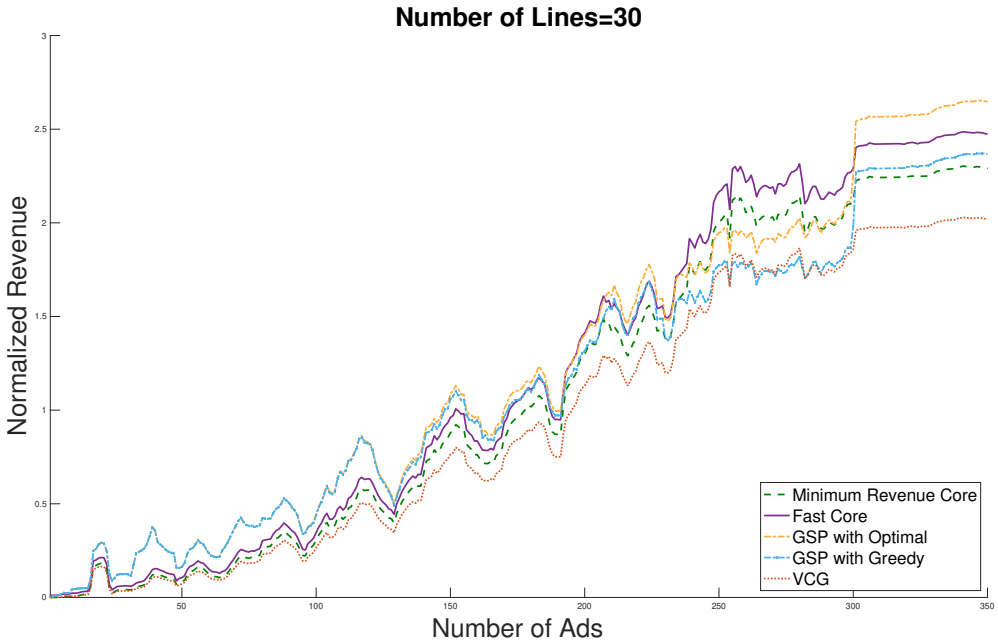

Figure 8 Normalized revenues versus total number of ads for line count=30.

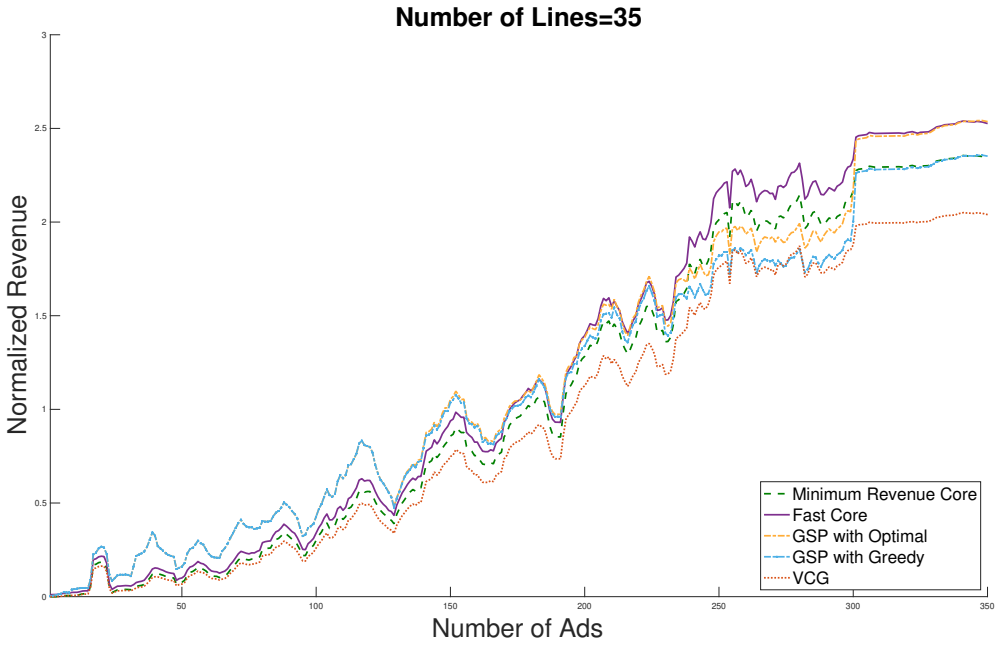

Figure 9 Normalized revenues versus total number of ads for line count=35.

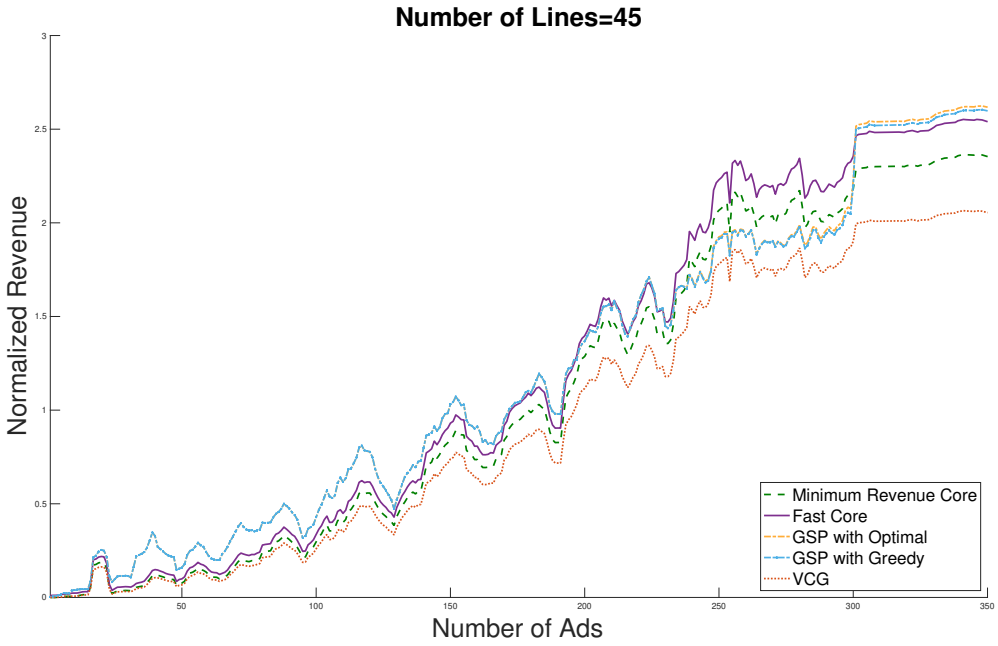

Figure 10 Normalized revenues versus total number of ads for line count=45.

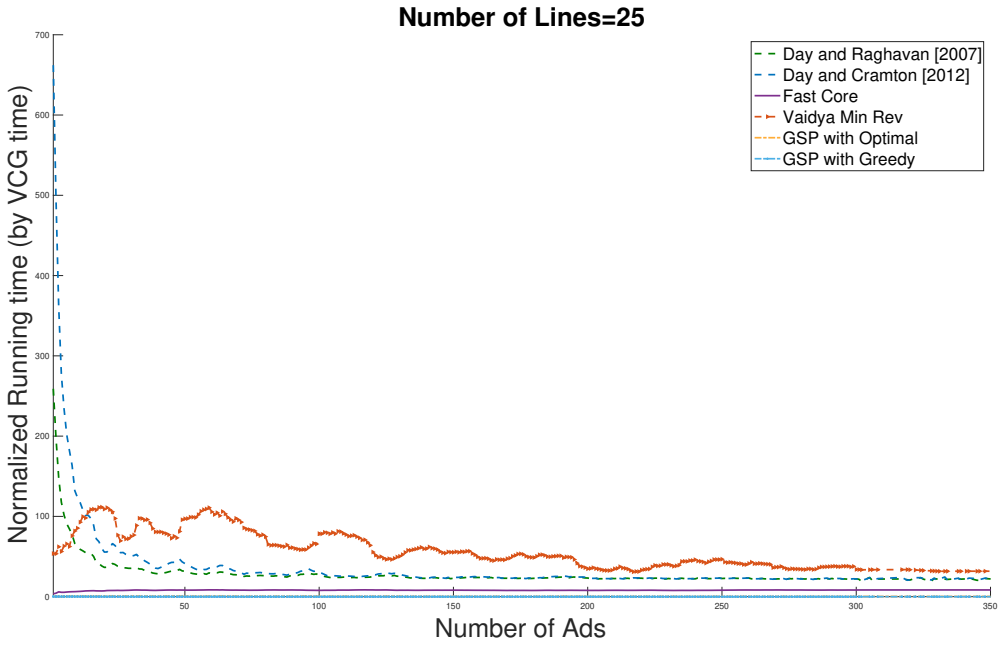

Figure 11 Normalized running times versus total number of ads for line count=25.

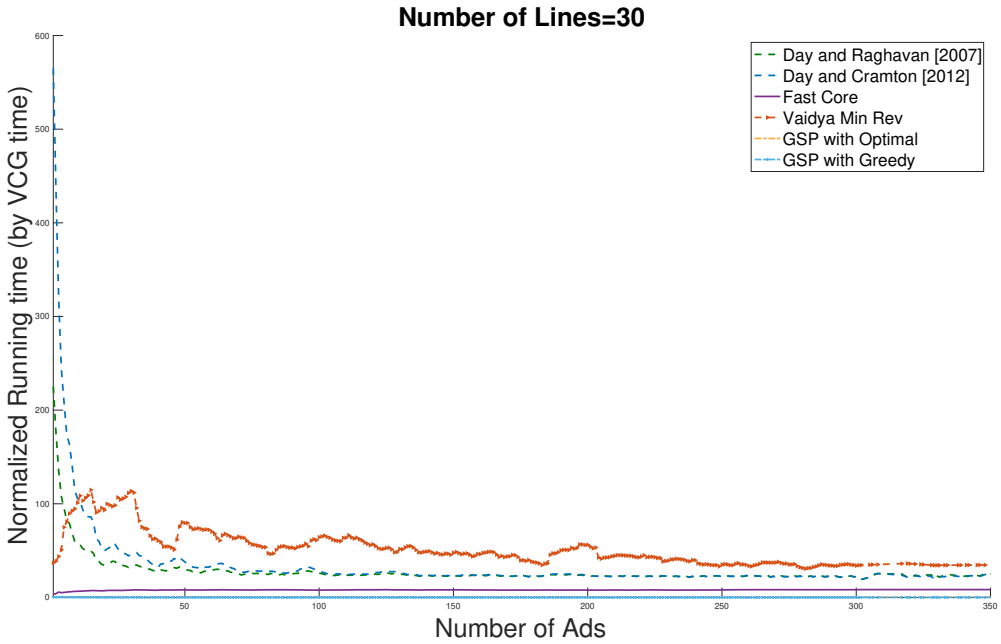

Figure 12 Normalized running times versus total number of ads for line count=30.

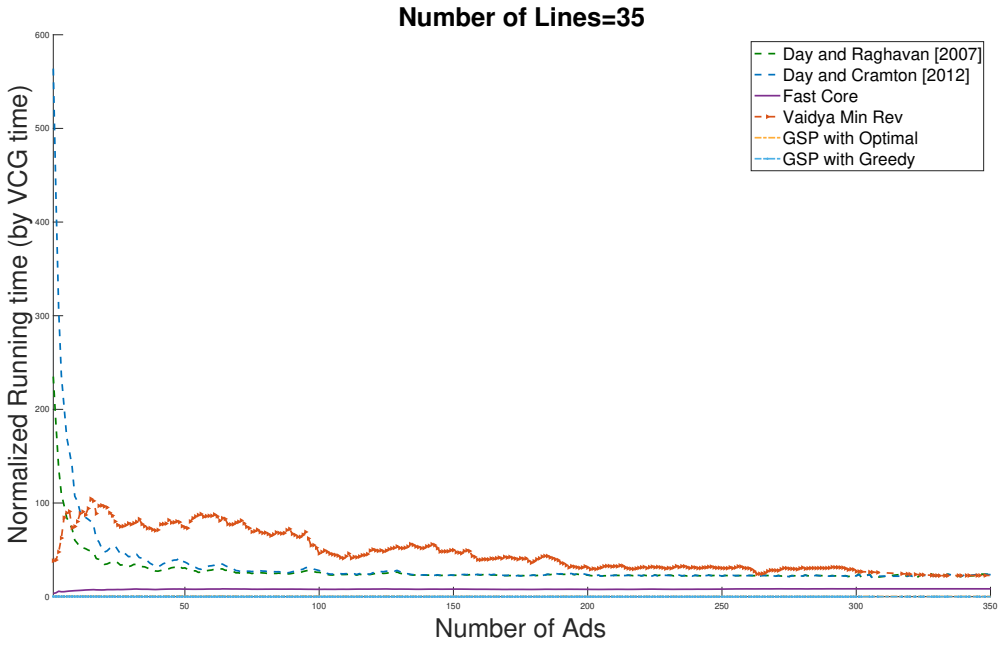

Figure 13 Normalized running times versus total number of ads for line count=35.

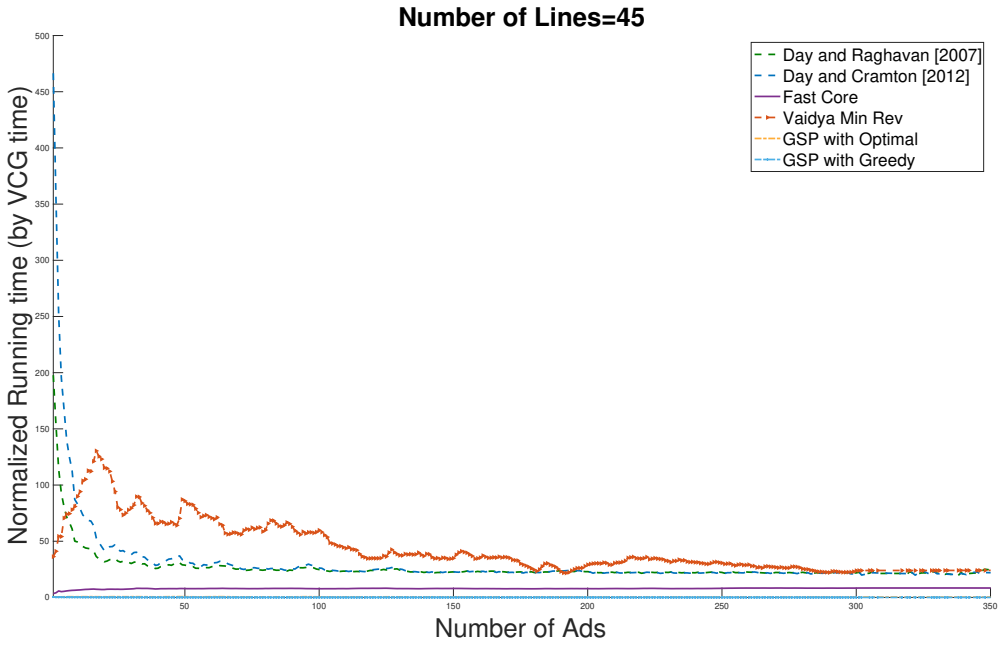

Figure 14 Normalized running times versus total number of ads for line count=45.

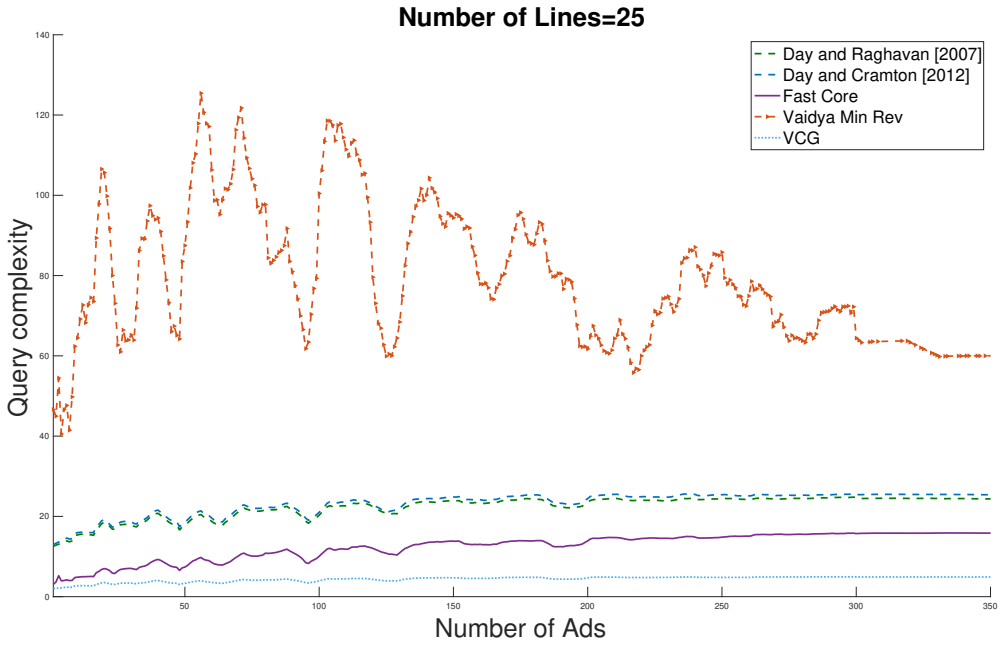

Figure 15 Query complexity versus total number of ads for line count=25.

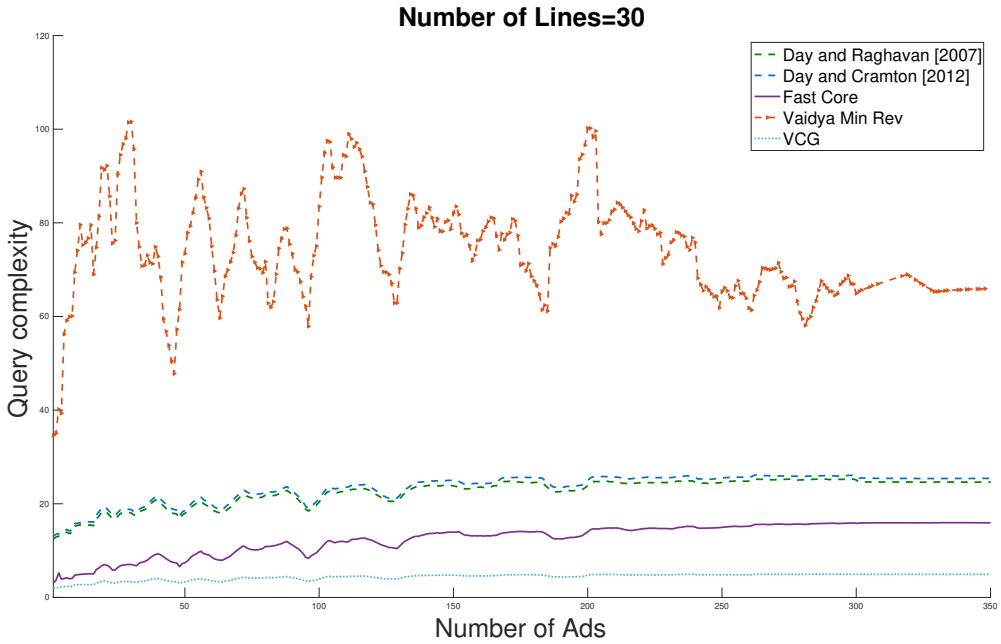

Figure 16 Query complexity versus total number of ads for line count=30.

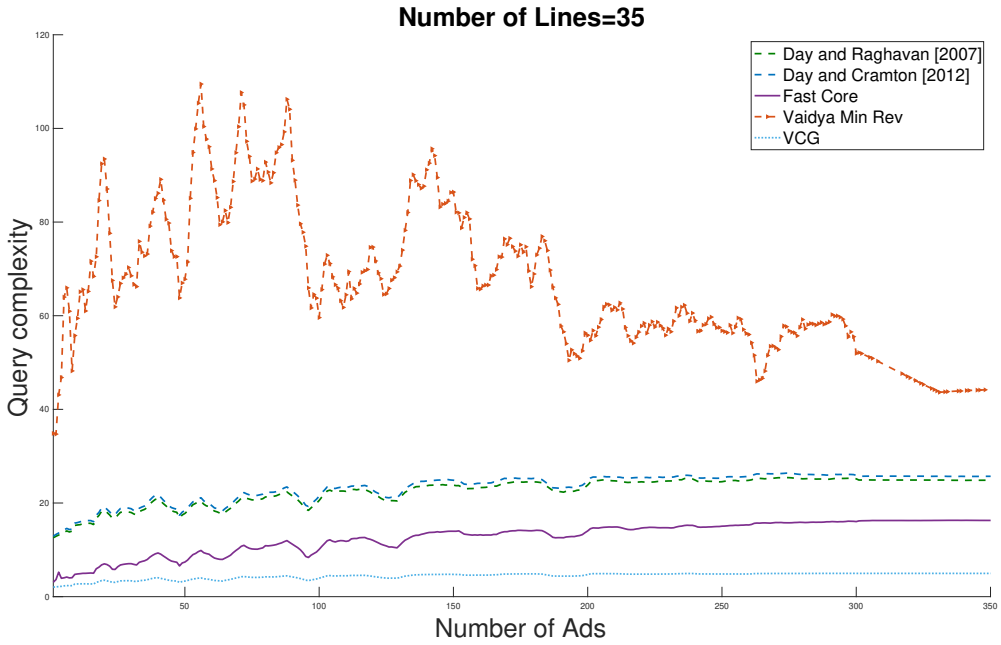

Figure 17 Query complexity versus total number of ads for line count=35.

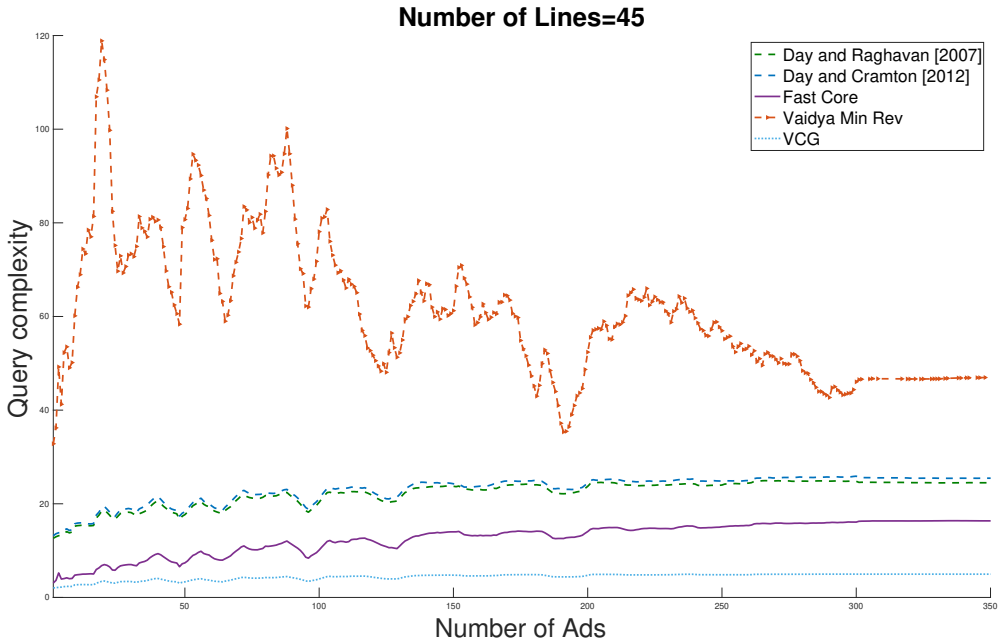

Figure 18 Query complexity versus total number of ads for line count=45.

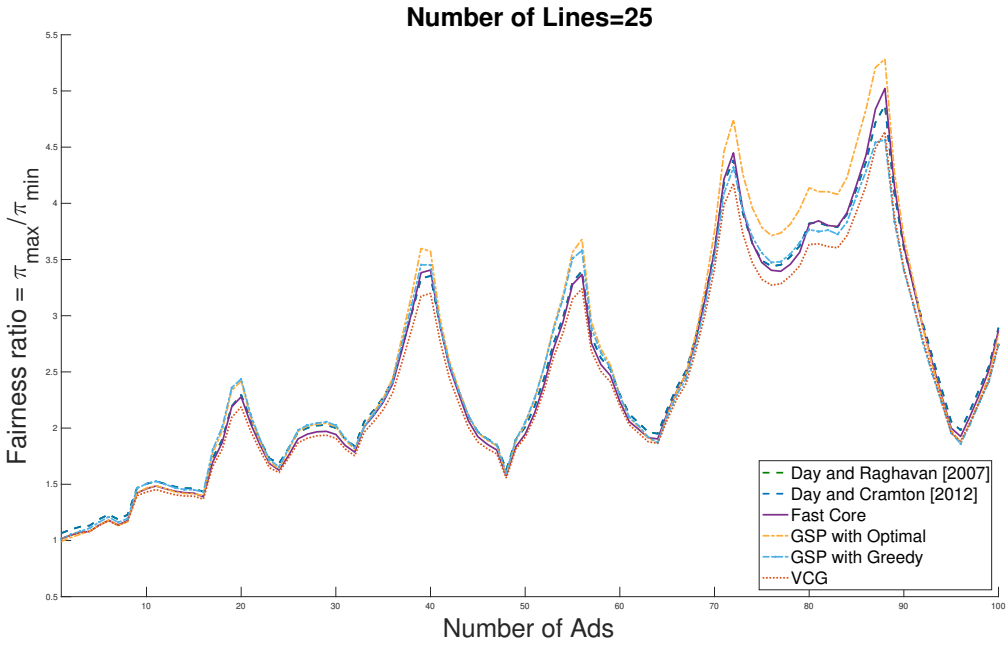

Figure 19 Fairness ratio versus total number of ads for line count=25.

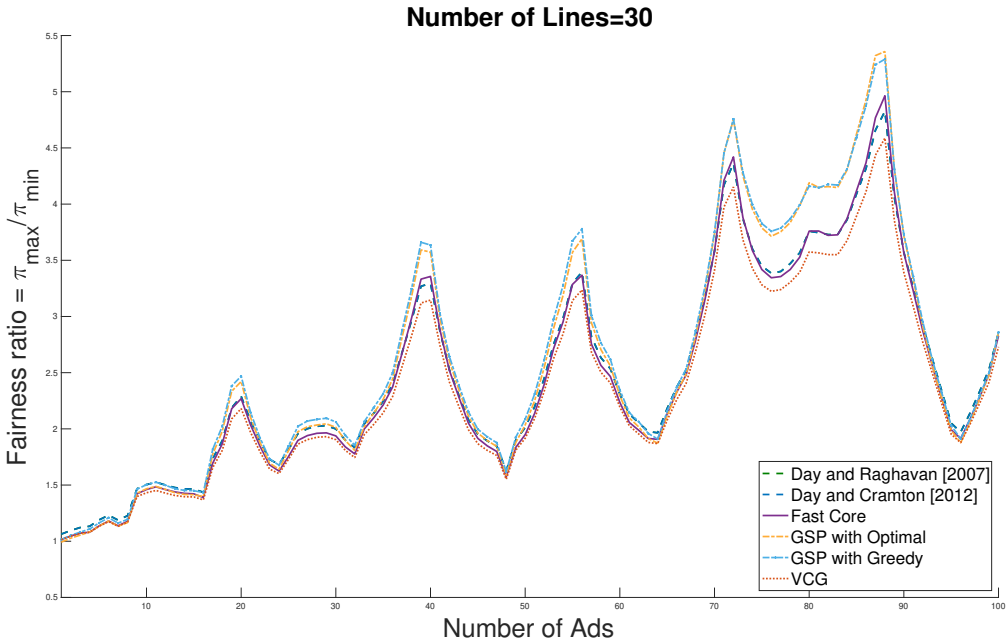

Figure 20 Fairness ratio versus total number of ads for line count=30.

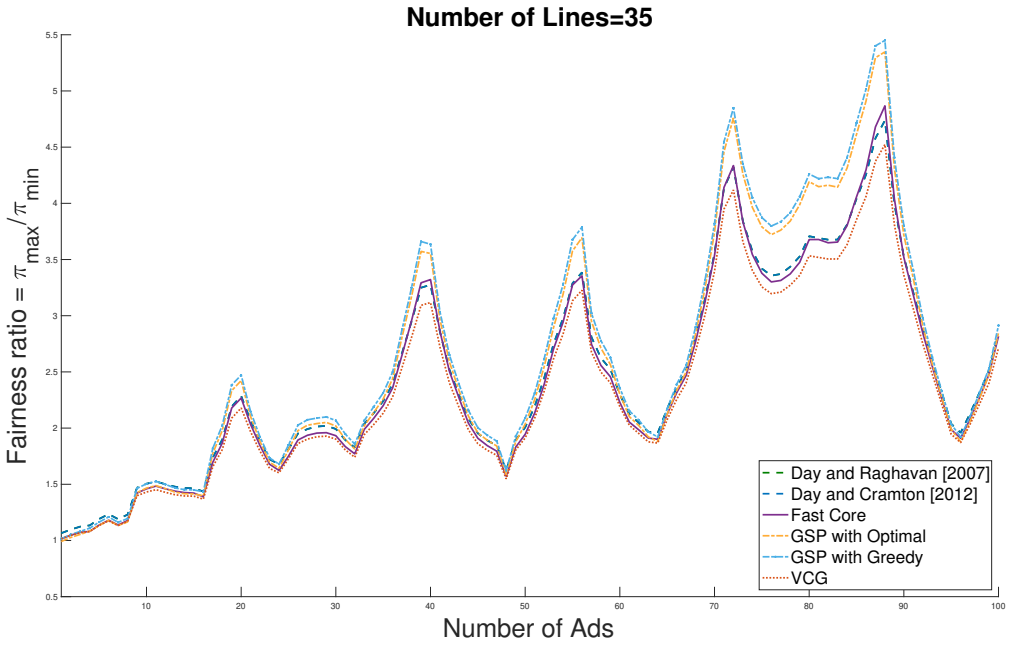

Figure 21 Fairness ratio versus total number of ads for line count=35.

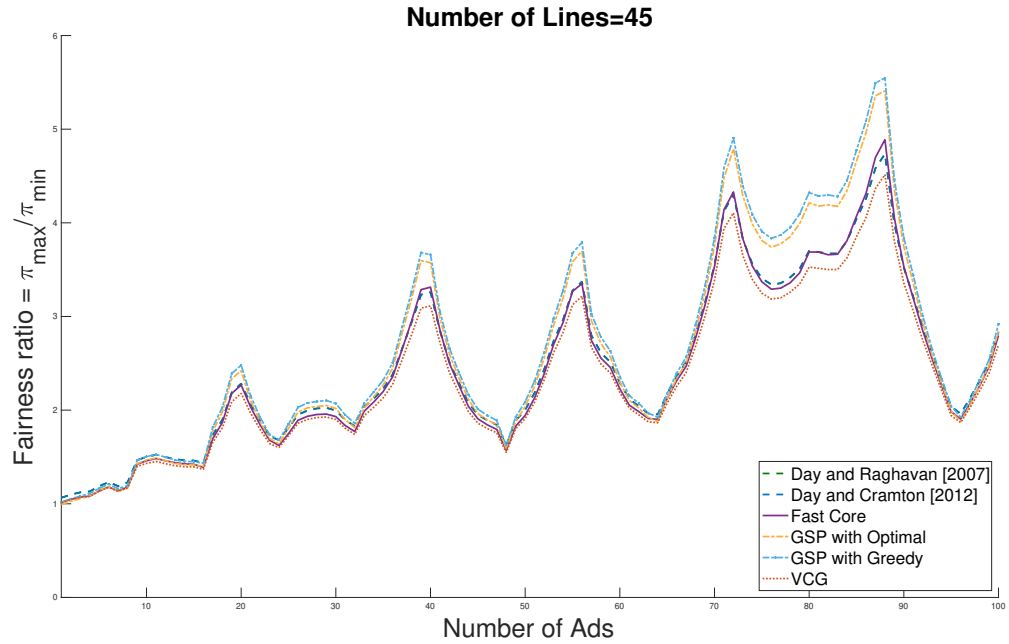

Figure 22 Fairness ratio versus total number of ads for line count=45.