Human review packet: EOS07GSP

Generated from byte-pinned source excerpts, the expanded PaperInterface specifications, and hash-bound raw-source-to-Spec screening records.

Paper: EOS07GSP
Paper id: EOS07GSP
Source version: NBER w11765; byte-pinned EOS07GSP.txt source extraction
Source record: audit/paper_statement_map.json
Rows: 13
Generated: 2026-09-06
Source URL: official source

How to use this packet

For each source claim, compare the exact source excerpts with the expanded PaperInterface specification. Mark up this PDF or fill the blank reviewer box in the TeX source; return the annotated file for reconciliation into the paper-local human-review log.

Open the interactive dashboard

The interactive dashboard is an optional alternative to using this PDF; it presents the same review surface rather than an additional review requirement. After forking and cloning (or pulling) AppliedModelingLib, run the following from the repository root. The cache command is needed only the first time in a new clone.

```
cd <your-repository-checkout>
lake exe cache get
python3 scripts/review_dashboard.py --paper EOS07GSP --serve
```

Open http://127.0.0.1:8765/ in a browser and keep that terminal running while reviewing. The dashboard records saved annotations in this paper's local reviewer-owned review record.

Contents

Source claims

Review row 1

Source locator: cited publication, lines 443-464

Verbatim source input

Definition 4 An equilibrium of the static game induced by GSP is locally envy-free if a player cannot improve his payoff by exchanging bids with the player ranked one position above him. More formally, in a locally envy-free equilibrium, for any i ≤ min{N, K}, we have αi sg(i) − p(i) ≥ αi−1 sg(i) − p(i−1) .
Of course, it is possible that in the dynamic game bids change over time, depending on the player's strategies and information structure. However, as long as the restrictions are satisfied, if the dynamic game ever converges to a static vector of bids, that static equilibrium should correspond to a "locally envy-free" equilibrium of the static game Γ induced by the GSP. Consequently, we view a locally envy-free equilibrium Γ as a prediction regarding a rest point at which the vector of bids stabilizes. In this section, we study the set of locally envy-free equilibria.
We first show that the set of locally envy-free equilibria maps naturally to a set of stable assignments in a corresponding two-sided matching market. The idea that auctions and two-sided matching models are closely related is not new: it goes back to Crawford and Knoer (1981), Kelso and Crawford (1982), Leonard (1983), and Demange, Gale, and Sotomayor (1986), and has been studied in detail in a recent paper by Hatfield and Milgrom (2005). Note, however, that in our case the non-standard auction is very different from those in the above papers.
Our environment maps naturally into the most basic assignment model, studied first by Shapley and Shubik (1972). We view each position as an agent who is looking for a match with an advertiser. The value of an advertiser-position pair (i, k) is equal to αk si . We call this assignment game A.
The advertiser makes its payment pik for the position, and the advertiser is left with αk si − pik .
The following pair of lemmas shows that there is a natural mapping from the set of locally envy-free equilibria of GSP to the set of stable assignments. All proofs are in the Appendix.

Formalized review target The archival source above is not asserted equivalent to this formalized target.

```
Corrected Definition 4 retains the printed adjacent allocated-rank conditions and additionally requires every unassigned bidder to weakly prefer remaining
↪    unassigned to taking the bottom allocated slot at its recorded GSP payment.
```

Archival source anchor: cited publication, lines 443-446

Expanded PaperInterface specification

```
∀ {Bidder : Type u_1} {Slot : Type u_2} [inst : DecidableEq Bidder]
  (E : AppliedModelingLib.Auction.PositionEnvironment Slot)
  (M : AppliedModelingLib.Auction.PositionMechanism Bidder Slot) (values bids : Bidder → ℝ) (allocatedPositions : ℕ)
  (bidderAtRank : ℕ → Bidder) (slotAtRank : ℕ → Slot),
  EOS07GSP.PaperInterface.correctedDefinition4LocallyEnvyFree E M values bids allocatedPositions bidderAtRank
      slotAtRank ↔
    EOS07GSP.PaperInterface.sourceDefinition4LocallyEnvyFree E M values bids allocatedPositions bidderAtRank
        slotAtRank ∧
      ∀ (bidder : Bidder),
        (M bids).slotOf bidder = none →
          ∀ (rank : ℕ),
            rank + 1 = allocatedPositions →
              E.clickThroughRate (slotAtRank rank) * (values bidder - (M bids).paymentPerClick (bidderAtRank rank)) ≤ 0
```

Lean proof endpoint (built separately)

```
EOS07GSP.PaperInterface.corrected_definition4_locally_envy_free
```

Recorded source-to-Spec screening **Verdict:** matches

Reason: The expanded target adds the all-unassigned-bidder bottom-slot condition and expressly does not present that addition as archival source text.

Reviewer check: ☐ Matches formalized target

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 2

Source locator: cited publication, lines 459-462

Verbatim source input

Our environment maps naturally into the most basic assignment model, studied first by Shapley and Shubik (1972). We view each position as an agent who is looking for a match with an advertiser. The value of an advertiser-position pair (i, k) is equal to αk si . We call this assignment game A. The advertiser makes its payment pik for the position, and the advertiser is left with αk si − pik .

Expanded PaperInterface specification

```
∀ {Bidder : Type u_1} {Slot : Type u_2} (E : AppliedModelingLib.Auction.PositionEnvironment Slot)
  (O : AppliedModelingLib.Auction.PositionOutcome Bidder Slot) (values : Bidder → ℝ) (i : Bidder) (s : Slot),
  O.slotOf i = some s →
    AppliedModelingLib.Auction.PositionOutcome.utility E O values i =
      E.clickThroughRate s * values i - E.clickThroughRate s * O.paymentPerClick i
```

Lean proof endpoint (built separately)

EOS07GSP.PaperInterface.stable_assignment

Recorded source-to-Spec screening **Verdict:** matches

Reason: The target uses the source assignment-game total-payment convention p_ik = alpha_k times the per-click price and utility alpha_k s_i minus p_ik.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 3

Source locator: cited publication, lines 239-248

Verbatim source input

Example. Suppose there are two slots on a page and three bidders. An ad in the first slot receives 200 clicks per hour, while the second slot gets 100. Bidders 1, 2, and 3 have values per click of $10, $4, and $2, respectively. Suppose bidder 2 bids $2.01, to guarantee that he gets a slot. Then bidder 1 will not want to bid more than $2.02—he does not need to pay more than that to get the top spot. But then bidder 2 will want to revise his bid to $2.03 to get the top spot, bidder 1 will in turn raise his bid to $2.04, and so on. Clearly, there is no pure strategy equilibrium in the one-shot version of the game, and so if bidders best respond to each other, they will want to revise their bids as often as possible. Figure 1 shows an example of this behavior on Overture.
2.2.3

Expanded PaperInterface specification

```
200 * (10 - 203 / 100) < 200 * (10 - 202 / 100) ∧
  100 * (4 - 201 / 100) < 200 * (4 - 203 / 100) ∧ 100 * (10 - 202 / 100) < 200 * (10 - 204 / 100)
```

Lean proof endpoint (built separately)

EOS07GSP.PaperInterface.first_price_running_example_profitable_revision_chain

Recorded source-to-Spec screening **Verdict:** matches

Reason: The three displayed inequalities formalize the source running example's profitable 2.02, 2.03, and 2.04 bid revisions with its 200/100 click rates and 10/4 values.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 4

Source locator: cited publication, lines 372-387

Verbatim source input

Remark 1 If all advertisers were to bid the same amounts under the two mechanisms, then each
advertiser's payment would be at least as large under GSP as under VCG.
10
In actual sponsored search auctions at Google and Yahoo!, advertisers can also choose to place "broad match"
bids that match searches that include a keyword along with additional search terms.
11
The actual practice at Overture is to show equal bids according to the order in which the bidders placed their
bids.
12
Although we normalize the reserve price to zero, search engines charge the last bidder a non-zero reserve price.

8

[form-feed in source extraction]
This is easy to show by induction on advertisers' payments, starting with the last advertiser
who gets assigned a position. For i = min{K, N }, p(i) = pV,(i) = αi b(i+1) . For any i < min{K, N },
pV,(i) − pV,(i+1) = (αi − αi+1)b(i+1) ≤ αi b(i+1) − αi+1 b(i+2) = p(i) − p(i+1) .

Expanded PaperInterface specification

```
∀ {value clickThroughRate : ℕ → ℝ},
  (∀ (i : ℕ), 0 ≤ value i) →
    (∀ (i : ℕ), value (i + 1) ≤ value i) →
      (∀ (i : ℕ), 0 ≤ clickThroughRate i) →
        ∀ (rank remaining : ℕ),
          AppliedModelingLib.Auction.paper_theorem7_ranked_vcg_tail_payment value clickThroughRate rank remaining ≤
            clickThroughRate rank * value (rank + 1)
```

Lean proof endpoint (built separately)

EOS07GSP.PaperInterface.remark1_gsp_payments_weakly_dominate_vcg

Recorded source-to-Spec screening **Verdict:** matches

Reason: The expanded target is the recursive VCG total-payment upper bound by the same-ranked truthful GSP total payment stated in Remark 1.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 5

Source locator: cited publication, lines 388-389

Verbatim source input

Remark 2 Truth-telling is a dominant strategy under VCG.
This is a well-known property of the VCG mechanism.

Expanded PaperInterface specification

```
∀ {Bidder : Type u_1} {Slot : Type u_2} [inst : Fintype Bidder] [inst_1 : DecidableEq Bidder] [inst_2 : Fintype Slot]
  [inst_3 : DecidableEq Slot] {E : AppliedModelingLib.Auction.PositionEnvironment Slot},
  (∀ (s : Slot), 0 < E.clickThroughRate s) →
    AppliedModelingLib.Auction.PositionMechanism.TruthfulDominantStrategy E
      (AppliedModelingLib.Auction.PositionMechanism.positionVCGMechanism E)
```

Lean proof endpoint (built separately)

EOS07GSP.PaperInterface.remark2_vcg_truthful

Recorded source-to-Spec screening **Verdict:** matches

Reason: The finite position-VCG target states the source Remark 2 conclusion that truthful bidding is a dominant strategy under VCG.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 6

Source locator: cited publication, lines 390-442

Verbatim source input

Remark 3 Truth-telling is not a dominant strategy under GSP.
For instance, consider a slight modification of the example from Section 2. There are still
three bidders, with values per click of \$10, \$4, and \$2, and two positions. However, the clickthrough rates of these positions are now almost the same: the
↪ first position receives 200 clicks per
hour, and the second one gets 199. If all players bid truthfully, then bidder 1's payoff is equal to
(\$10 − \$4) ∗ 200 = \$1200. If, instead, he shades his bid and bids only \$3 per click, he will get the
second position, and his payoff will be equal to (\$10 − \$2) ∗ 199 = \$1592 > \$1200.

4

Static GSP and Locally Envy-Free Equilibria13

Advertisers bidding on Yahoo! and Google can change their bids very frequently. We therefore
think of these sponsored search auctions as continuous time or infinitely repeated games in which
bidders originally have private information about their types, gradually learn the values of others,
and can adjust their bids repeatedly. In principle, the sets of equilibria in such repeated games can
be very large, with players potentially punishing each other for deviations. However, the strategies
required to support such equilibria are usually quite complex, requiring precise knowledge of the
environment and careful implementation. It may not be reasonable to expect the advertisers to
be able to execute such strategies: they often manage thousands of keywords, and implementing
sophisticated dynamic strategies for many keywords is likely to be prohibitively expensive and
complex. In theory, advertisers could implement such strategies via automated robots, but at
present they cannot do so: bidding software must first be authorized by the search engines,14 and
search engines are unlikely to permit sophisticated strategies that would allow bidders to collude
and reduce revenues. Indeed, leading software for bidding at sponsored search auctions does not
allow advertisers to condition their bids on the prior behavior of specific competitors.15
We therefore focus on simple strategies that bidders can reasonably execute, and we study the
rest points of the bidding process: if the vector of bids stabilizes, at what bids can it stabilize?
We impose several assumptions and restrictions. First, we assume that all values are common
knowledge: over time, bidders are likely to learn all relevant information about each other's values.
13
After this research was completed, we found out that several of the results of this section were independently
discovered by Hal Varian in unpublished work.
14
See, e.g., http://www.content.overture.com/d/HKm/legal/zhhkrtc.jhtml (accessed August 18, 2005).
15
See, e.g., Atlas Onepoint Rules-Based Bidding, http://www.atlasonepoint.com/products/bidmanager/rulesbased
(accessed August 18, 2005), which lacks any bidding rule of this type.

9

[form-feed in source extraction]
Second, since bids can be changed at any time, stable bids must be static best responses to each
other—otherwise a bidder whose bid is not a best response would have an incentive to change
it. Thus, we assume that the bids form an equilibrium in the static one-shot game of complete
information. Third, what are the simple strategies that a bidder can use to increase his payoff,
beyond simple best responses to the other players' bids?
One clear strategy is to try to force out the player who occupies the position immediately above.
Suppose bidder k bids bk and is assigned to position i + 1 and bidder k 0 bids bk0 > bk and is assigned
to position i. Note that if k raises his bid slightly, his own payoff does not change, but the payoff
of the player above him decreases. Of course, player k 0 can retaliate, and the most he can do is to
slightly underbid bidder k, effectively swapping places with him. If bidder k is better off after such
retaliation, he will indeed want to force player k 0 out, and the vector of bids will change. Thus,
if the vector converges to a rest point, an advertiser in position k should not want to "exchange"
positions with the advertiser in position k − 1. We call such vectors of bids "locally envy-free."

Expanded PaperInterface specification

```
AppliedModelingLib.Auction.PositionMechanism.utility EOS07GSP.remark3SourceEnvironment
     AppliedModelingLib.Auction.gsp3TwoSlotMechanism EOS07GSP.remark3SourceValues EOS07GSP.remark3SourceValues 0 =
   1200 ∧
  AppliedModelingLib.Auction.PositionMechanism.utility EOS07GSP.remark3SourceEnvironment
       AppliedModelingLib.Auction.gsp3TwoSlotMechanism EOS07GSP.remark3SourceValues EOS07GSP.remark3SourceShadedBids
       0 =
     1592 ∧
   ¬AppliedModelingLib.Auction.PositionMechanism.TruthfulDominantStrategy EOS07GSP.remark3SourceEnvironment
       AppliedModelingLib.Auction.gsp3TwoSlotMechanism
```

Lean proof endpoint (built separately)

EOS07GSP.PaperInterface.remark3_gsp_not_truthful

Recorded source-to-Spec screening **Verdict:** matches

Reason: The target preserves the source two-slot 200/199, values 10/4/2 shading witness, including truthful utility 1200, shaded utility 1592, and failure of truthfulness as a dominant strategy.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 7

Source locator: cited publication, lines 239-278

Verbatim source input

Example. Suppose there are two slots on a page and three bidders. An ad in the first slot receives 200 clicks per hour, while the second slot gets 100. Bidders 1, 2, and 3 have values per click of $10, $4, and $2, respectively. Suppose bidder 2 bids $2.01, to guarantee that he gets a slot. Then bidder 1 will not want to bid more than $2.02—he does not need to pay more than that to get the top spot. But then bidder 2 will want to revise his bid to $2.03 to get the top spot, bidder 1 will in turn raise his bid to $2.04, and so on. Clearly, there is no pure strategy equilibrium in the one-shot version of the game, and so if bidders best respond to each other, they will want to revise their bids as often as possible. Figure 1 shows an example of this behavior on Overture.
2.2.3

Generalized Second-Price Auctions

Under the generalized first-price auction, the bidder who could react to its competitors' moves fastest had a substantial advantage. The mechanism therefore encouraged inefficient investments in gaming the system. It also created volatile prices that in turn caused allocative inefficiencies. Google addressed these problems when it introduced its own pay-per-click system, AdWords Select, in February 2002. Google also recognized that a bidder in position i will never want to pay more than one bid increment above the bid of the advertiser in position (i + 1), and Google adopted this principle in its newly-designed generalized second price auction mechanism. In the simplest GSP auction, an advertiser in position i pays a price per click equal to the bid of an advertiser in position (i + 1) plus a minimum increment (typically $0.01). This second-price structure makes the market more user friendly and less susceptible to gaming.
Recognizing these advantages, Yahoo!/Overture also switched to GSP. Let us describe the version of GSP that it implemented.5 Every advertiser submits a bid. Advertisers are arranged on the page in descending order of their bids. The advertiser in the first position pays a price per click that equals the bid of the second advertiser plus an increment; the second advertiser pays the price offered by the third advertiser plus an increment; and so forth.6
Example (continued). Let us now consider the payments in the environment of the previous example under GSP mechanism. If all advertisers bid truthfully, then bids are $10, $4, $2. Payments in GSP will be $4 and $2.7 Truth-telling is indeed an equilibrium in this example, because no bidder can benefit by changing his bid. Note that total payments of bidders one and two are $800 and $200, respectively.
5

We focus on Overture's implementation, because Google's system is somewhat more complex. However, it is straightforward to generalize our analysis to Google's mechanism.
6
For this version of GSP to be efficient, it is necessary that the number of clicks received by an advertisement in a given position depends on the ad's position and not on the advertiser's identity. Recognizing that this assumption may not hold if some ads attract more clicks than others, Google adjusts effective bids based on ads' click-through

Expanded PaperInterface specification

AppliedModelingLib.Auction.PositionMechanism.IsNashEquilibrium
  AppliedModelingLib.Auction.paper_eos_running_example_environment AppliedModelingLib.Auction.gsp3TwoSlotMechanism
  AppliedModelingLib.Auction.paper_eos_running_example_values3
  AppliedModelingLib.Auction.paper_eos_running_example_values3

Lean proof endpoint (built separately)

EOS07GSP.PaperInterface.running_example_truthful_gsp_nash

Recorded source-to-Spec screening **Verdict:** matches

Reason: The finite two-slot, three-bidder target states the source 200/100 and 10/4/2 truthful-GSP equilibrium example.
Reviewer check: ☐ Matches source input
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

Review row 8

Source locator: cited publication, lines 266-315

Verbatim source input

Example (continued). Let us now consider the payments in the environment of the previous
example under GSP mechanism. If all advertisers bid truthfully, then bids are $10, $4, $2. Payments
in GSP will be $4 and $2.7 Truth-telling is indeed an equilibrium in this example, because no bidder
can benefit by changing his bid. Note that total payments of bidders one and two are $800 and
$200, respectively.
5

We focus on Overture's implementation, because Google's system is somewhat more complex. However, it is
straightforward to generalize our analysis to Google's mechanism.
6
For this version of GSP to be efficient, it is necessary that the number of clicks received by an advertisement in
a given position depends on the ad's position and not on the advertiser's identity. Recognizing that this assumption
may not hold if some ads attract more clicks than others, Google adjusts effective bids based on ads' click-through
rates. But under the assumption that all ads have the same click-through rates conditional on position, Google's
and Yahoo!'s versions of GSP are identical. With modification for Google's adjustment procedure, our results for the
Yahoo! version of GSP also hold for Google adjusted GSP.
7
For convenience, we neglect the $0.01 minimum increments.

6

[form-feed in source extraction]
2.3

Generalized Second-Price and VCG Auctions

GSP looks similar to the Vickrey-Clarke-Groves mechanism, because both mechanisms set each
agent's payments only based on the allocation and bids of other players, not based on that agent's
own bid. In fact, Google's advertising materials explicitly refer to Vickrey and state that Google's
"unique auction model uses Nobel Prize-winning economic theory to eliminate ... that feeling that
you've paid too much".8 But GSP is not VCG. In particular, unlike the VCG auction, GSP does
not have an equilibrium in dominant strategies, and truth-telling is generally not an equilibrium
strategy in GSP. (See the example in Remark 3.) With only one slot, VCG and GSP would be
identical. With several slots, the mechanisms are different: while GSP essentially charges bidder i
the bid of bidder i + 1, VCG charges bidder i the externality that he imposes on others, i.e., the
decrease in the value of clicks received by other bidders because of i's presence.
Example (continued). Let us compute VCG payments for the example considered above.
The second bidder's payment is $200, as before. However, the payment of the first advertiser is now
$600: $200 for the externality that he imposes on bidder 3 (by forcing him out of position 2) and
$400 for the externality that he imposes on bidder 2 (by moving him from position 1 to position
2 and thus causing him to lose (200 − 100) = 100 clicks per hour). Note that in this example,
revenues under VCG are lower than under GSP. As we will show later, if advertisers were to bid
their true values under both mechanisms, revenues would always be higher under GSP.

2.4

Assessing the Market's Development

The chronology above suggests three major stages in the development of the sponsored search
advertising market. First, ads were sold manually, slowly, in large batches, and on a cost-perimpression basis. Second, Overture implemented
↪ keyword-targeted per-click sales and began to
streamline advertisement sales with some self-serve bidding interfaces, but with a highly unstable

Expanded PaperInterface specification

```
AppliedModelingLib.Auction.paper_eos_running_example_value 1 = 4 ∧
 AppliedModelingLib.Auction.paper_eos_running_example_value 2 = 2 ∧
  AppliedModelingLib.Auction.paper_eos_running_example_clickThroughRate 0 *
     AppliedModelingLib.Auction.paper_eos_running_example_value 1 =
    800 ∧
   AppliedModelingLib.Auction.paper_eos_running_example_clickThroughRate 1 *
      AppliedModelingLib.Auction.paper_eos_running_example_value 2 =
     200 ∧
    AppliedModelingLib.Auction.paper_theorem7_ranked_vcg_tail_payment
       AppliedModelingLib.Auction.paper_eos_running_example_value
       AppliedModelingLib.Auction.paper_eos_running_example_clickThroughRate 0 2 =
      600 ∧
     AppliedModelingLib.Auction.paper_theorem7_ranked_vcg_tail_payment
        AppliedModelingLib.Auction.paper_eos_running_example_value
        AppliedModelingLib.Auction.paper_eos_running_example_clickThroughRate 1 1 =
       200 ∧
      AppliedModelingLib.Auction.paper_eos_running_example_clickThroughRate 0 *
         AppliedModelingLib.Auction.paper_eos_running_example_value 1 +
        AppliedModelingLib.Auction.paper_eos_running_example_clickThroughRate 1 *
         AppliedModelingLib.Auction.paper_eos_running_example_value 2 >
       AppliedModelingLib.Auction.paper_theorem7_ranked_vcg_tail_payment
         AppliedModelingLib.Auction.paper_eos_running_example_value
         AppliedModelingLib.Auction.paper_eos_running_example_clickThroughRate 0 2 +
        AppliedModelingLib.Auction.paper_theorem7_ranked_vcg_tail_payment
         AppliedModelingLib.Auction.paper_eos_running_example_value
         AppliedModelingLib.Auction.paper_eos_running_example_clickThroughRate 1 1
```

Lean proof endpoint (built separately)

EOS07GSP.PaperInterface.running_example_truthful_gsp_revenue_comparison

Recorded source-to-Spec screening **Verdict:** matches
Reason: The target explicitly records the source GSP per-click prices 4 and 2, GSP totals 800 and 200, VCG totals 600 and 200, and the resulting strict revenue comparison 1000 > 800.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 9

Source locator: cited publication, lines 463-466

Verbatim source input

The following pair of lemmas shows that there is a natural mapping from the set of locally envy-free equilibria of GSP to the set of stable assignments. All proofs are in the Appendix.
Lemma 5 The outcome of any locally envy-free equilibrium of auction Γ is a stable assignment.
Lemma 6 If the number of bidders is greater than the number of available positions, then any

[Next verbatim source excerpt]

Proof of Lemma 5
By definition, in any locally envy-free equilibrium outcome, no player can profitably re-match with the position assigned to the player right above him. Also, no player (a) can profitably re-match with a position assigned to a player below him (b)—if such a profitable re-matching existed, player a would find it profitable to slightly undercut player b in game Γ and get b's position and payment. But this would contradict the assumption that we are in equilibrium.22
Hence, we only need to show that no player can profitably re-match with the position assigned to a player more than one spot above him. First, note that in any locally envy-free equilibrium, the resulting matching must be assortative, i.e., for any i, the player assigned to position i has a higher per-click valuation than the player assigned to position i + 1, and therefore, player 1 must be assigned to the top position, player 2—to the second-highest position, and so on.
Indeed, let si and si+1 denote the values of players assigned to positions i and i+1. Equilibrium restrictions imply that αi si − pi ≥ αi+1 si − pi+1 (nobody wants to move one position down), and local envy-freeness implies that αi+1 si+1 − pi+1 ≥ αi si+1 − pi (nobody wants to move one position up). Manipulating the above inequalities yields αi si − αi si+1 + αi+1 si+1 ≥ αi+1 si , thus (αi − αi+1)si ≥ (αi − αi+1)si+1 . Since αi > αi+1 , we have si ≥ si+1 , and hence the locally envy-free equilibrium outcome must be an assortative match.
Now, let us show that no player can profitably re-match with the position assigned to a player more than one spot above him. Suppose bidder k is considering re-matching with position m < k−1. Since the equilibrium is locally envy-free, we have
αk sk − pk ≥ αk−1 sk − pk−1
αk−1 sk−1 − pk−1 ≥ αk−2 sk−1 − pk−2
..
.
αm+1 sm+1 − pm+1 ≥ αm sm+1 − pm .
Since αj > αj+1 for any j, and sj > sk for any j < k, the above inequalities remain valid after replacing sj with sk . Doing that, then adding all inequalities up, and canceling out the redundant elements, we get αk sk − pk ≥ αm sk − pm . But that implies that advertiser k cannot re-match profitably with position m, and we are done.

Formalized review target

The archival source above is not asserted equivalent to this formalized target.

On the finite strict ranked-GSP domain, the corrected Definition 4 implies that the ordinary sorted-GSP outcome is a stable assignment.

Archival source anchor: cited publication, lines 463-466

Expanded PaperInterface specification

```
∀ {m n : ℕ} (hn : 0 < n) (hnm : n < m) {value clickThroughRate : ℕ → ℝ} (bids : Fin m → ℝ),
  (∀ {i j : Fin m}, ↑i < ↑j → bids j < bids i) →
    (∀ (i : Fin n), 0 < clickThroughRate ↑i) →
      (∀ (k : ℕ), k + 1 < n → clickThroughRate (k + 1) < clickThroughRate k) →
        (EOS07GSP.PaperInterface.correctedDefinition4LocallyEnvyFree
            (AppliedModelingLib.Auction.paper_theorem7_ranked_environment clickThroughRate)
            (AppliedModelingLib.Auction.paper_ranked_gsp_mechanism m n) (fun (i : Fin m) => value ↑i) bids n
            (fun (rank : ℕ) => if h : rank < n then ⟨rank, Nat.lt_trans h hnm⟩ else ⟨0, Nat.lt_trans hn hnm⟩)
            fun (rank : ℕ) => if h : rank < n then ⟨rank, h⟩ else ⟨0, hn⟩) →
          AppliedModelingLib.Auction.PositionOutcome.StableAssignment
            (AppliedModelingLib.Auction.paper_theorem7_ranked_environment clickThroughRate)
            (AppliedModelingLib.Auction.paper_ranked_gsp_mechanism m n bids) fun (i : Fin m) => value ↑i
```

Lean proof endpoint (built separately)

EOS07GSP.PaperInterface.corrected_lemma5_ranked_more_bidders_stable

Recorded source-to-Spec screening

Verdict: matches
Reason: The full finite K>N route states the authorized corrected Definition 4 condition and derives the source stable-assignment conclusion on that disclosed domain.
Reviewer check: ☐ Matches formalized target
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

Review row 10

Source locator: cited publication, lines 466-480

Verbatim source input

Lemma 6 If the number of bidders is greater than the number of available positions, then any
stable assignment is an outcome of a locally envy-free equilibrium of auction Γ.
10

[form-feed in source extraction]
We will now construct a particular locally envy-free equilibrium of game Γ. This equilibrium
has two important properties. First, in this equilibrium bidders' payments coincide with their
payments in the dominant- strategy equilibrium of VCG. Second, this equilibrium is the worst
locally envy-free equilibrium for the search engine, and it is the best locally envy-free equilibrium
for the bidders. Consequently, the revenues of a search engine are (weakly) higher in any locally
envy-free equilibrium of GSP than in the dominant-strategy equilibrium of VCG.
Consider the following strategy profile B ∗ . For each advertiser i ∈ {2, . . . , min{N, K}}, bid b∗i
V,(i−1)

is equal to p αi−1 , where pV,(j) is the payment of bidder j in the dominant-strategy equilibrium of
VCG. Bid b∗1 is equal to s1 .16

[Next verbatim source excerpt]

Proof of Lemma 6
Take a stable assignment. By a result of Shapley and Shubik (1972), this assignment must be
efficient, and hence assortative, and so we can talk about advertiser i being matched with position
22

This argument relies on the fact that in equilibrium, no two (or more) players bid the same amount, which is
straightforward to prove: since all players' per-click values are different, and all ties are broken randomly with equal
probabilities, at least one of such players would find it profitable to bid slightly higher or slightly lower.

15

[form-feed in source extraction]
i, with associated payment pi .
Let us construct a locally envy-free equilibrium with the corresponding outcome. Let b1 = s1
and bi = αpi−1
for i > 1. Let us show that this set of strategies is a locally envy-free equilibrium.
i−1

First, note that for any i, bi > bi+1 (because otherwise we would have, for some i, αpi−1
≤ αpii ⇒
i−1

si − αpi−1
≥ si − αpii ⇒ αi−1 (si − pi−1) > αi (si − pi), which would imply that player i could rei−1
match profitably). Therefore, position allocations and payments resulting from this strategy profile
will coincide with those in the original stable assignment. To see that this strategy profile is an
equilibrium, note that deviating and moving to a different position in this strategy profile is at
most as profitable for any player as re-matching with the corresponding position in the assignment
game. To see that this equilibrium is locally envy-free, note that the payoff from swapping with
the bidder above is exactly equal to the payoff from re-matching with that player's position in the
assignment game.

Formalized review target The archival source above is not asserted equivalent to this formalized target.

On the finite K>N ranked domain, a stable assignment with strictly positive utility for every assigned bidder has the strict-surplus GSP implementation and
↪ locally-envy-free equilibrium route, preserving every assignment and each assigned bidder's payment.

Archival source anchor: cited publication, lines 466-467

Expanded PaperInterface specification

```
∀ {m n : ℕ},
  0 < n →
    ∀ (hnm : n < m) {value payment clickThroughRate : ℕ → ℝ}
      (O : AppliedModelingLib.Auction.PositionOutcome (Fin m) (Fin n)),
      EOS07GSP.PaperInterface.correctedLemma6StableAssignment
          (AppliedModelingLib.Auction.paper_theorem7_ranked_environment clickThroughRate) (fun (i : Fin m) => value ↑i)
          O →
        (∀ (i : Fin n), O.slotOf ⟨↑i, Nat.lt_trans i.isLt hnm⟩ = some i) →
          (∀ (bidder : Fin m), n ≤ ↑bidder → O.slotOf bidder = none) →
            (∀ (i : Fin n), O.paymentPerClick ⟨↑i, Nat.lt_trans i.isLt hnm⟩ = payment ↑i / clickThroughRate ↑i) →
              (∀ (i : Fin n), 0 < clickThroughRate ↑i) →
                (∀ (k : ℕ), k + 1 < n → clickThroughRate (k + 1) < clickThroughRate k) →
                  have bids : Fin m → ℝ :=
                    EOS07GSP.PaperInterface.lemma6ConstructedBidsMoreBidders value payment clickThroughRate;
                  EOS07GSP.PaperInterface.lemma6AssignedOutcomeEq
                      (AppliedModelingLib.Auction.paper_ranked_gsp_tiebreak_mechanism m n bids) O ∧
                    AppliedModelingLib.Auction.PositionMechanism.LocallyEnvyFreeEquilibrium
                      (AppliedModelingLib.Auction.paper_theorem7_ranked_environment clickThroughRate)
                      (AppliedModelingLib.Auction.paper_ranked_gsp_tiebreak_mechanism m n) (fun (i : Fin m) => value ↑i)
                      bids
```

Lean proof endpoint (built separately)

EOS07GSP.PaperInterface.corrected_lemma6_ranked_more_bidders_constructs_tiebreak_equilibrium

Recorded source-to-Spec screening **Verdict:** matches

Reason: The target makes the strict-surplus and assigned-payment-only outcome repair explicit before proving the finite construction and locally envy-free equilibrium.

Reviewer check: ☐ Matches formalized target

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 11

Source locator: cited publication, lines 470-483

Verbatim source input

We will now construct a particular locally envy-free equilibrium of game Γ. This equilibrium has two important properties. First, in this equilibrium bidders' payments coincide with their payments in the dominant- strategy equilibrium of VCG. Second, this equilibrium is the worst locally envy-free equilibrium for the search engine, and it is the best locally envy-free equilibrium for the bidders. Consequently, the revenues of a search engine are (weakly) higher in any locally envy-free equilibrium of GSP than in the dominant-strategy equilibrium of VCG.
Consider the following strategy profile B ∗ . For each advertiser i ∈ {2, . . . , min{N, K}}, bid b∗i
V,(i−1)

is equal to p αi−1 , where pV,(j) is the payment of bidder j in the dominant-strategy equilibrium of VCG. Bid b∗1 is equal to s1 .16
Theorem 7 Strategy profile B ∗ is a locally envy-free equilibrium of game Γ. In this equilibrium, each bidder's position and payment is equal to those in the dominant-strategy equilibrium of the game induced by VCG. In any other locally envy-free equilibrium of game Γ, the total revenue of

[Next verbatim source excerpt]

Proof of Theorem 7
First, we need to check that the order of the bids is preserved, i.e., b∗i > b∗i+1 for any i < min{N, K}.
For i ≥ 2, this is equivalent to

pV,(i−1)
pV,(i)
>
αi−1
αi
m
(αi−1 − αi)si + pV,(i)
pV,(i)
>
αi−1
αi
m
αi (αi−1 − αi)si > (αi−1 − αi)pV,(i)
m
αi si > pV,(i) .

For i = 1, b∗i > b∗i+1 is equivalent to
s1 >

pV,(1)
α1

m
α1 s1 > pV,(1) .

To see that for any i, αi si > pV,(i) , note first that in the game induced by VCG, each player can guarantee himself the payoff of at least zero (by bidding zero), and hence in any equilibrium his
16

[form-feed in source extraction]
payoff from clicks is at least as high as his payment. To prove that the inequality is strict, note that if player i's value per click were slightly lower, e.g., si − ∆ instead of si , ∆ < si − si+1 , then his payment in the truth-telling equilibrium would still be the same (because it does not depend on his own bid, given the allocation of positions), and so pV,(i) ≤ αi (si − ∆) < αi si . Thus, for any i, b∗i > b∗i+1 , and therefore each bidder's position is the same as in the truthful equilibrium of VCG. Therefore, by construction, payments are also the same.
Next, to see that no bidder i can benefit by bidding less than b∗i , suppose that he bids an amount b0 < b∗i that puts him in position i0 > i. Then, by construction, his payment will be equal to the amount that he would need to pay to be in position i0 under VCG, provided that other players bid truthfully. But truthful bidding is an equilibrium under VCG, and so such deviation cannot be profitable there—hence, it cannot be profitable in strategy profile B ∗ of game Γ either.
To see that no bidder i can benefit by bidding more than b∗i , suppose that he bids an amount b0 > b∗i that puts him in position i0 < i. Then the net payoff from this deviation is equal to
Pi−1 V,(j)
Pi−1
−
(αi0 −αi)si −(αi0 b∗i0 −αi bi+1) < (αi0 −αi)si −(αi0 b∗i0 +1 −αi bi+1) = j=i
0 (αj −αj+1)si −
j=i0 (p
Pi−1
Pi−1
V,(j+1)
p
) = j=i0 (αj − αj+1)si − j=i0 (αj − αj+1)sj+1 . But since si ≤ sj+1 for any j < i, the last expression is less than or equal to zero, and hence the deviation is not profitable.
To check that this equilibrium is locally envy-free, note that if bidder i swapped his bids with bidder i−1, his payoff would change by (αi−1 −αi)si −(αi−1 b∗i −αi bi+1) = (αi−1 −αi)si −(pV,(i−1) − pV,(i)) = (αi−1 − αi)si − (pV,(i−1) − pV,(i)) = (αi−1 − αi)si − (αi−1 − αi)si = 0. In other words, each bidder is indifferent between his actual payoff and his payoff after swapping bids with the bidder

Expanded PaperInterface specification

```
∀ (value vcgTotalPayment clickThroughRate : ℕ → ℝ) (i : ℕ),
  clickThroughRate i ≠ 0 →
    clickThroughRate i *
```

```
  AppliedModelingLib.Auction.paper_theorem7_bstar_bid value vcgTotalPayment clickThroughRate (i + 1) =
vcgTotalPayment i
```

Lean proof endpoint (built separately)

EOS07GSP.PaperInterface.theorem7_bstar_payment_identity

Recorded source-to-Spec screening **Verdict:** matches

Reason: The target is the B-star next-price identity alpha_i b^*_{i+1} = p_i^V used in the source proof's VCG payment ledger.

Reviewer check: ☐ Matches source input

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation

Review row 12

Source locator: cited publication, lines 481-490

Verbatim source input

Theorem 7 Strategy profile B ∗ is a locally envy-free equilibrium of game Γ. In this equilibrium, each bidder's position and payment is equal to those in the dominant-strategy equilibrium of the game induced by VCG. In any other locally envy-free equilibrium of game Γ, the total revenue of the seller is at least as high as in B ∗ .
To prove Theorem 7, we first note that payments under strategy profile B ∗ coincide with VCG payments and check that B ∗ is indeed a locally-envy free equilibrium. This follows from the fact that, by construction, each bidder is indifferent between remaining in his positions and swapping with the bidder one position above him. Next, from Lemma 5 we know that every locally envy-free equilibrium corresponds to a stable assignment. Combining this with the fact17 that in any stable assignment the payments of bidders must be at least as high as VCG payments completes the proof.

[Next verbatim source excerpt]

locally envy-free equilibrium for the search engine. A standard result of matching theory states that there exists an assignment that is the best stable assignment for all advertisers, and the worst stable assignment for all positions. A stable assignment is characterized by a vector of payments p = (p1 , . . . , pK). Let pV = (pV1 , . . . , pVK) be the set of dominant-strategy VCG payments, i.e., the set of payments in equilibrium B ∗ of game Γ.
In any stable assignment, pK must be at least as high as αK sK−1 , since otherwise advertiser K − 1 would find it profitable to match with position K. On the other hand, pVK = αK sK−1 , and hence in the buyer-optimal stable assignment, pK = pVK .
Next, in any stable assignment, it must be the case that pK−1 − pK ≥ (αK−1 − αK)sK — otherwise, advertiser K would find it profitable to re-match with position K − 1. Hence, pK−1 ≥ (αK−1 −αK)sK +pK ≥ (αK−1 −αK)sK +pVK = pVK−1 , and so in the buyer-optimal stable assignment, pK−1 = pVK−1 .
Proceeding by induction, we get pk = pVk for any k ≤ K in the buyer-optimal (and therefore seller-pessimal) stable assignment, and so in any locally envy-free equilibrium of game Γ, the total
P
V
revenue of the seller is at least as high as K

Formalized review target   The archival source above is not asserted equivalent to this formalized target.

On the finite strict ranked-GSP comparison domain, the corrected Definition 4 yields a B-star locally-envy-free GSP equilibrium with VCG-equivalent ↪ positions and payments, whose actual seller revenue is weakly below every corrected comparison profile.

Archival source anchor: cited publication, lines 481-490

Expanded PaperInterface specification

```
∀ {m n : ℕ} (hnm : n < m) (model : EOS07GSP.EOSFiniteStaticBStarOrder n) (bids : Fin m → ℝ)
  (hstrict : ∀ {i j : Fin m}, ↑i < ↑j → bids j < bids i)
  (hcorrected :
    EOS07GSP.PaperInterface.correctedDefinition4LocallyEnvyFree
      (AppliedModelingLib.Auction.paper_theorem7_ranked_environment model.clickThroughRate)
      (AppliedModelingLib.Auction.paper_ranked_gsp_mechanism m n) (fun (i : Fin m) => model.value ↑i) bids n
      (fun (rank : ℕ) =>
        if h : rank < n then ⟨rank, Nat.lt_trans h hnm⟩ else ⟨0, Nat.lt_trans model.slots_nonempty hnm⟩)
      fun (rank : ℕ) => if h : rank < n then ⟨rank, h⟩ else ⟨0, model.slots_nonempty⟩),
  EOS07GSP.PaperInterface.theorem7_finite_static_bstar_tiebreak_conclusionSpec model ∧
    EOS07GSP.PaperInterface.theorem7_finite_static_bstar_revenue_minimal_correctedSpec hnm model bids
      (fun {i j : Fin m} => hstrict) hcorrected
```

Lean proof endpoint (built separately)

EOS07GSP.PaperInterface.theorem7_finite_static_bstar_full_corrected

Recorded source-to-Spec screening   **Verdict:** matches

Reason: The direct finite strict-ranked GSP route proves B-star's VCG ledger and seller-minimal revenue under the explicitly corrected Definition 4 domain.
Reviewer check: ☐ Matches formalized target
If left unchecked, explain the discrepancy or uncertainty below.
Reviewer annotation

Review row 13

Source locator: cited publication, lines 528-590

Verbatim source input

K 6= N + 1 require only minor modifications in the proof.) Click-through rates αn are commonly
known, with αN +1 ≡ 0. Bidders' per-click valuations si are drawn from a continuous distribution
F (·) on [0; +∞) with a continuous density function f (·) that is positive everywhere on (0, +∞).
Each advertiser knows his valuation and the distribution of other advertisers' valuations.
The strategy of an advertiser assigns the choice of dropping out or not for any history of the
game, given that the advertiser has not previously dropped out. In other words, the strategy can be
represented as a function pi (k, h, si), where si is the value per click of bidder i, pi is the price at which
he drops out, k is the number of bidders remaining (including bidder i), and h = (bk+1 , . . . , bK) is
the history of prices at which bidders K, K − 1, . . . , k + 1 have dropped out. (As a result, the
price that bidder i would have to pay per click if he dropped out next is equal to bk+1 , unless the
history is empty, in which case we say that bk+1 ≡ 0.)
Theorem 8 In the unique perfect Bayesian equilibrium of the generalized English auction with
k
strategies continuous in si , an advertiser with value si drops out at price pi (k, h, si) = si − ααk−1
(si −

bk+1). In this equilibrium, each advertiser's resulting position and payoff are the same as in the
dominant-strategy equilibrium of the game induced by VCG. This equilibrium is ex post: the strategy
of each bidder is a best response to other bidders' strategies regardless of their realized values.
The intuition of the proof is as follows. First, with k players remaining and the next highest
bid equal to bk+1 , it is a dominated strategy for a player with value s to drop out before price p
reaches the level at which he is indifferent between getting position k and paying bk+1 per click and
getting position k − 1 and paying p per click. Next, if for some set of types it is not optimal to drop
out at this "borderline" price level, we can consider the lowest such type, and then once the clock
reaches this price level, a player of this type will know that he has the lowest per-click value of the
18

If several players drop out simultaneously, one of them is chosen randomly. Whenever a player drops out, the
clock is stopped, and other players are also allowed to drop out; again, if several ones want to drop out, one is chosen
randomly. If several players end up dropping at the same price, the first one to drop out is placed in the lowest
position of the still available ones, the next one—to the position right above that, and so on.
19
Without this restriction, multiple equilibria exist, even in the simplest English auction with two bidders and
one object. For example, suppose there is one object for sale and two bidders with independent private values for
this object distributed exponentially on [0, ∞). Consider the following pair of strategies. If a bidder's value is in
the interval [0, 1] or in the interval [2, ∞), he drops out when the clock reaches his value. If bidder 1's value is in
the interval (1, 2), he drops out at 1, and if bidder 2's value is in the interval (1, 2), he drops out at 2. This pair of
strategies, together with appropriate beliefs, forms a perfect Bayesian equilibrium.

12

[form-feed in source extraction]
remaining players. But then he will also know that the other remaining players will only drop out
at price levels at which he will find it unprofitable to compete with them for the higher positions.
The result of Theorem 8 resembles the classic result on the equivalence of the English auction
and the second price sealed-bid auction under private values (Vickrey, 1961). Note, however, that
the intuition is very different: Vickrey's result follows simply from the existence of equilibria in
dominant strategies, whereas in our case such strategies do not exist, and bids do depend on other
player's bids. Also, our result is very different from the revenue equivalence theorem: payoffs in the
generalized English auction coincide with VCG payments for all realizations of values, not only in
expectation; bidders can be asymmetric; distributions of valuations need not be commonly known.
The equilibrium described in Theorem 8 is an ex post equilibrium. As long as all bidders other
than bidder i follow the equilibrium strategy described in Theorem 8, it is a best response for bidder
i to follow his equilibrium strategy, for any realization of other bidders' values. Thus, the outcome
implemented by this mechanism depends only on the realization of bidders' values and does not
depend on bidders' beliefs about each others' types.
Obviously, any dominant strategy solvable game has an ex post equilibrium. However, the
generalized English auction is not dominant strategy solvable. This combination of properties is
quite striking: the equilibrium is unique and efficient, the strategy of each bidder does not depend
on the distribution of other bidders' values, yet bidders do not have dominant strategies.20,21 The
generalized English auction is a particularly interesting example, because it can be viewed as a
model of a mechanism that has "emerged in the wild."

6

[Next verbatim source excerpt]

[form-feed in source extraction]
Proof of Theorem 8
First, note that in equilibrium, for any player i, any history h, and any number of remaining players
k, the drop-out price pi (k, h, si) tends to infinity as si tends to infinity. (Otherwise there would
exist a player for whom it was optimal to deviate from his strategy and stay longer, for a sufficiently
high value s.) Next, take any equilibrium of the generalized English auction. Note that if in this
equilibrium pi (k, h, si) > pi (k, h, s0i) for some k, h, i, and types si < s0i , then it has to be the
case that both types si and s0i are indifferent between dropping out at pi (k, h, si) and pi (k, h, s0i).
(Otherwise one of them would be able to increase his payoff by mimicking the other.) Consequently,
we can "swap" such players' strategies, and therefore there exists an "observationally equivalent"
equilibrium in which strategies are nondecreasing in types; also, they are still continuous in own
values. Consider this equilibrium profile of strategies pi (k, h, si).
Let q(k, bk+1 , s) be such a price that a player with value s is indifferent between getting position
k at price bk+1 and position k − 1 at price q(k, bk+1 , s). That is,
αk−1 (s − q(k, bk+1 , s)) = αk (s − bk+1)
m
q(k, bk+1 , s) = s −

αk
(s − bk+1).
αk−1

Slightly abusing notation, let q(k, h, s) = q(k, bk+1 , s), where bk+1 is the last bid at which a player
dropped out in history h. (This player received position k + 1. If history h is empty, we set
bk+1 = 0.) We will now show that for any i, k, h, and si , pi (k, h, si) = q(k, h, si).
Suppose that is not the case, and take the largest k for which there exist such history h (with
the last player dropping out at bk+1), player i, and type si (surviving with positive probability on
the equilibrium path) that pi (k, h, si) 6= q(k, h, si). Since by assumption, all strategies up to this
stage were pi (·, ·, ·) = q(·, ·, ·), we know that there exists a value smin ≥ bk+1 , such that all players
with values less than smin have dropped out, and all players with values greater than smin are still
in the auction.
Step 1. Suppose for some type s ≥ smin , pmax (k, h, s) ≡ maxi pi (k, h, s) > q(k, h, s). Let s0
be the smallest type, and let i be the corresponding player, such that pi (k, h, s0) = pmax (k, h, s).
Without loss of generality, we can assume that there is a positive mass of types of other players
dropping out at or before pi (k, h, s0).23
Step 1(a). Suppose first that there is a positive mass of types of other players dropping out
at pi (k, h, s0) = pmax (k, h, s). That implies that with positive probability, player i of type s0 will
remain in the subgame following the drop-out of some other player at pi (k, h, s0) (since ties are
broken randomly). Let us show that in this subgame, player i of type s0 will be the first player to
23

Otherwise, we have ∀j 6= i, pj (k, h, s0) ≤ pi (k, h, s) ≤ pmax (k, h, s) = pi (k, h, s0) ⇒ ∀j 6= i, pj (k, h, s0) =
pi (k, h, s0) and ∀s0 > s0 , pj (k, h, s0) > pi (k, h, s0). But we also have pi (k, h, s0) = pmax (k, h, s) > q(k, h, s) ≥
q(k, h, s0), and so for some s0 > s0 we have pmax (k, h, s0) > q(k, h, s0) and pmax (k, h, s0) > pmax (k, h, s). We
can then consider s00 in place of s0 , where s00 is the smallest type, and i0 is the corresponding player, such that
pi0 (k, h, s00) = pmax (k, h, s0). There is a positive mass of types of other players dropping out before pi0 (k, h, s00).

18

[form-feed in source extraction]
drop out with probability 1. Suppose that is not the case, and let l < k − 1 be the smallest number
such that he gets position l with positive probability.
Consider any continuation of history h, hl+2 , such that the last player to drop out in that history
gets position l+2 and drops out at price bl+2 , player i of type s0 is one of the remaining l+1 players,
there is a positive probability that player i gets position l in the continuation subgame following
history hl+2 , and there is zero probability that player i gets position m for any m < l. Note
that s0 ≥ bl+2 —otherwise it would have been optimal for player i to drop out earlier. Consider
pi (l + 1, hl+2 , s0). There must be a positive mass of types of other players who drop out no
later than pi (l + 1, hl+2 , s0). Take the highest such type, s0 , and the corresponding player j.
It has to be the case that s0 > s0 ≥ bl+2 . It also has to be the case that q(l + 1, hl+2 , s0) is
less than or equal to pi (l + 1, hl+2 , s0). (Otherwise player j with value s0 would be playing a
strategy weakly dominated by dropping out at q(l + 1, hl+2 , s0), with a positive probability of
earning strictly less than he would have earned if he waited until that price level.) Therefore,
pi (l + 1, hl+2 , s0) ≥ q(l + 1, hl+2 , s0) > q(l + 1, hl+2 , s0) ≥ bl+2 . Let us show that it would be
strictly better for player i with type s0 to drop out at q(l + 1, hl+2 , s0) instead of waiting until
pi (l + 1, hl+2 , s0). Indeed, if nobody else drops out in between, or someone drops out before
q(l + 1, hl+2 , s0), these strategies would result in identical payoffs. Otherwise payoffs are different,
and this happens with positive probability. Under the former strategy, player i earns
αl+1 (s0 − bl+2).
Under the latter strategy, he earns
αl (s0 − bl+1),
where bl+1 is the price at which somebody else dropped out. (The probability of getting a spot
m < l is zero by construction.) With probability 1, bl+1 ≥ q(l + 1, hl+2 , s0), and with positive
probability bl+1 > q(l + 1, hl+2 , s0), so the expected payoff from waiting until pi (l + 1, hl+2 , s0) is
strictly less than the expected payoff from dropping out at q(l + 1, hl+2 , s0): E[αl (s0 − bl+1)] <
αl (s0 − q(l + 1, hl+2 , s0)) = αl (s0 − (s0 −

αl+1
αl (s0 − bl+2))) = αl+1 (s0 − bl+2).

Therefore, in the subgame following the drop-out of some other player at pi (k, h, s0), player i
of type s0 gets position k − 1 with probability 1, and therefore his payoff is αk−1 (s0 − pi (k, h, s0)).
Now suppose player i dropped out at a price pi (k, h, s0) − [U+000F control character] instead of waiting until pi (k, h, s0).
If somebody else drops out before pi (k, h, s0) − [U+000F control character] or after pi (k, h, s0), or drops out at pi (k, h, s0)
but player i is chosen to drop out first, then these two strategies result in identical payoffs. The
probability that somebody drops out in the interval (pi (k, h, s0) − [U+000F control character], pi (k, h, s0)) goes to zero as [U+000F control character]
goes to zero, and the possible difference in the payoffs is finite, so the difference in payoffs due
to this contingency goes to zero as [U+000F control character] goes to zero. Finally, there is a positive probability that
somebody else drops out at pi (k, h, s0) and is chosen to drop out first. If player i drops out before
that, at pi (k, h, s0) − [U+000F control character], his payoff is αk (s0 − bk+1). If he waits until pi (k, h, s0), we know that in the
subsequent subgame his payoff is αk−1 (s0 − pi (k, h, s0)) < αk−1 (s0 − q(k, h, s0)) = αk (s0 − bk+1).

19

[form-feed in source extraction]
Therefore, for a sufficiently small [U+000F control character], it is strictly better for player i with value s0 to drop out at
pi (k, h, s0) − [U+000F control character] instead of waiting until pi (k, h, s0), which contradicts the assumption that {pj (·, ·, ·)}
is an equilibrium.
Step 1(b). Now suppose there is mass zero of types of other players dropping out at pi (k, h, s0) =
pmax (k, h, s), but there is a positive mass dropping out before pi (k, h, s0). Consider a sequence of
small positive numbers ([U+000F control character]n) converging to 0 as n → ∞ and sequences (πn1), (πn2), . . . , (πnk−1), where
πnl is the probability that player i with value s0 will end up in position l if another player drops
out at price (pi (k, h, s0) − [U+000F control character]n). Let B = α1 s0 , i.e., the maximum payoff that a player with value s0
can possibly get in the auction. Now, if (πnk−1) converges to 1 and (πnl) converges to zero for all
l < k − 1, then, by an argument similar to the one at the end of Step 1(a), it is better for player
i of type s0 to drop out at some time pi (k, h, s0) − [U+000F control character].24 If (πnk−1) does not converge to 1, take the
highest l for which (πnl) does not converge to zero, and take a subsequence of [U+000F control character]n along which (πnl)
converges to some positive number ρ. Let s1 be the value such that for a random draw of types of
remaining players other than i, conditional on each draw being greater than s0 , the probability that
Q
1−F (s)
at least one type is less than s1 is equal to ρ2 (i.e., j6=i 1−Fjj (s01) = 1 − ρ2). Clearly, s1 > s0 . Take a
small [U+000F control character]n , and consider a subgame following some continuation of history h, hl+2 , where the (l + 2)nd

player drops out at b_{l+2}, $l + 1$ players, including player i, remain, and player i gets position l with probability close to ρ (and any position higher than l with probability close to zero). Consider $p_i(l + 1, h_{l+2}, s_0)$. There must exist a player, j, such that $p_j(l + 1, h_{l+2}, s_1) \le p_i(l + 1, h_{l+2}, s_0)$. (Otherwise, the probability of player i surviving until position l is less than or close to ρ^2, and thus cannot be close to ρ.) But then, by an argument similar to the one in Step 1(a), in this subgame it is strictly better for player i to drop out slightly earlier than $p_i(l + 1, h_{l+2}, s_0)$: Conditional on somebody else dropping out in between, the benefit is close to zero (the probability of getting a position better than l times the highest possible benefit B), while the cost is close to a positive number (the payoff from being in position $l + 1$ at price b_{l+2} vs. the payoff from being in position l at price at least $s_1 -$

α_{l+1}
α_l $(s_1 - b_{l+2})$—contradiction.

Step 2. In Step 1, we have shown that $p_{max}(k, h, s) \equiv \max_i p_i(k, h, s)$ cannot be greater than $q(k, h, s)$, and therefore for any player i and type $s \ge s_{min}$, $p_i(k, h, s) \le q(k, h, s)$. Take some value $s > s_{min}$ and player i. Suppose $p_i(k, h, s) < q(k, h, s)$. Take some other player j. From Step 1, we have $p_j(k, h, s_{min}) \le q(k, h, s_{min}) < q(k, h, s)$, and therefore if player i waited until $q(k, h, s)$ instead of dropping out at $p_i(k, h, s)$, the probability that someone dropped out in between would be positive, and hence the payoff would be strictly greater (by the definition of function $q()$, player i with value s strictly prefers being in position $k - 1$ or higher at any price less than $q(k, h, s)$ to being in position k at price b_{k+1}), which is impossible in equilibrium. Hence, $p_i(k, h, s) = q(k, h, s)$ for all $s > s_{min}$. By continuity, we also have $p_i(k, h, s_{min}) = q(k, h, s_{min})$.

24

If some other player drops out between $p_i(k, h, s_0) -$ [U+000F control character] and $p_i(k, h, s_0)$, the benefit of staying longer tends to zero (it is at most $B(1 - \pi_{k-1}))$, while the cost converges to a positive number (the difference between getting position k at price b_{k+1} and position $k - 1$ at price $p_i(k, h, s_0)$).

Formalized review target

The archival source above is not asserted equivalent to this formalized target.

On the finite K=N+1 generalized-English domain, the corrected Theorem 8 target proves the named continuous symmetric continuation plan is an ex-post
↪ PBE with bidder-indexed induced histories and beliefs, effective action uniqueness among symmetric continuation plans, and the terminal GSP-to-VCG
↪ ledger.

Archival source anchor: cited publication, lines 528-546

Expanded PaperInterface specification

```
∀ {Bidder : Type u_1} [inst : Fintype Bidder] [inst_1 : DecidableEq Bidder] [inst_2 : Nonempty Bidder]
  (law : EOS07GSP.PaperInterface.Theorem8ContinuousValueLaw) (clickThroughRate : ℕ → ℝ) (values : Bidder → ℝ),
  1 < Fintype.card Bidder →
    (∀ (bidder : Bidder), 0 ≤ values bidder) →
      (∀ rank < Fintype.card Bidder - 1, 0 < clickThroughRate rank) →
        (∀ rank < Fintype.card Bidder - 1, 0 ≤ clickThroughRate (rank + 1)) →
          (∀ rank < Fintype.card Bidder - 1, clickThroughRate (rank + 1) < clickThroughRate rank) →
            clickThroughRate (Fintype.card Bidder - 1) = 0 →
              ∀ (defaultBidder : Bidder),
                have plan : EOS07GSP.PaperInterface.Theorem8ContinuationPlan :=
                  EOS07GSP.PaperInterface.theorem8NamedContinuationPlan clickThroughRate;
                have strategy : EOS07GSP.PaperInterface.Theorem8ContinuousHistoryStrategy Bidder :=
                  EOS07GSP.PaperInterface.theorem8ContinuousHistoryStrategy Bidder clickThroughRate;
                have finalState : EOS07GSP.PaperInterface.Theorem8StoppedSourceState Bidder :=
                  EOS07GSP.PaperInterface.theorem8StoppedSourceFinalState strategy values;
                let initialState : EOS07GSP.PaperInterface.Theorem8StoppedSourceState Bidder :=
                  EOS07GSP.PaperInterface.theorem8StoppedSourceInitialState Bidder;
                have hremaining : initialState.remaining.Nonempty := Finset.univ_nonempty;
                have rankedValue : ℕ → ℝ :=
                  EOS07GSP.PaperInterface.theorem8StoppedSourceTerminalRankedValue strategy values initialState
                    hremaining defaultBidder;
                EOS07GSP.PaperInterface.Theorem8FiniteLegalHistoryExPostPBE Bidder law clickThroughRate
                    (Fintype.card Bidder - 1) plan ∧
                  (∀ (otherPlan : EOS07GSP.PaperInterface.Theorem8ContinuationPlan),
                      EOS07GSP.PaperInterface.Theorem8FiniteLegalHistoryExPostPBE Bidder law clickThroughRate
                          (Fintype.card Bidder - 1) otherPlan →
                        ∀ (rank : ℕ) (history : EOS07GSP.PaperInterface.Theorem8SourcePriceHistory) (ownValue : ℝ),
                          rank < Fintype.card Bidder - 1 →
                            EOS07GSP.PaperInterface.theorem8SourcePriceHistoryLastDropout history ≤ ownValue →
                              max (EOS07GSP.PaperInterface.theorem8SourcePriceHistoryLastDropout history)
                                  (otherPlan rank history ownValue) =
                                AppliedModelingLib.Auction.paper_theorem8_generalized_english_indifference_price
                                  (clickThroughRate rank) (clickThroughRate (rank + 1))
                                  (EOS07GSP.PaperInterface.theorem8SourcePriceHistoryLastDropout history) ownValue) ∧
                    plan.historyStrategy Bidder = strategy ∧
                      (EOS07GSP.PaperInterface.theorem8StoppedSourceFinalRankedBidders strategy values).toFinset =
                          Finset.univ ∧
                        List.Pairwise (fun (earlier later : Bidder) => values later ≤ values earlier)
                            (EOS07GSP.PaperInterface.theorem8StoppedSourceFinalRankedBidders strategy values) ∧
                          (∀ bidder ∈ finalState.remaining,
                              finalState.terminalGSPOutcome.slotOf bidder = some 0 ∧
                                finalState.terminalGSPOutcome.paymentPerClick bidder =
                                  List.getD finalState.history 0 0) ∧
                            (∀ bidder ∉ finalState.remaining,
                                bidder ∈ finalState.dropped.dropLast →
                                  finalState.terminalGSPOutcome.slotOf bidder =
                                      some (List.idxOf bidder finalState.dropped + 1) ∧
                                    finalState.terminalGSPOutcome.paymentPerClick bidder =
                                      List.getD finalState.history (List.idxOf bidder finalState.dropped + 1) 0) ∧
                              (∀ bidder ∉ finalState.remaining,
                                  bidder ∉ finalState.dropped.dropLast →
                                    finalState.terminalGSPOutcome.slotOf bidder = none) ∧
```

```
∀ index < List.length finalState.history,
  clickThroughRate index * List.getD finalState.history index 0 =
    AppliedModelingLib.Auction.paper_theorem7_ranked_vcg_tail_payment rankedValue
      clickThroughRate index (finalState.dropped.length - index)
```

Lean proof endpoint (built separately)

```
EOS07GSP.PaperInterface.theorem8_finite_legal_history_pbe_terminal_vcg_ledger
```

Recorded source-to-Spec screening **Verdict:** matches

Reason: The archival source uses bidder-indexed p_i(k,h,s_i), while this explicitly target quantifies one common symmetric continuation plan, induces bidder-indexed histories and beliefs from it, and restricts uniqueness to that common-plan class.

Reviewer check: ☐ Matches formalized target

If left unchecked, explain the discrepancy or uncertainty below.

Reviewer annotation