CSC304: Algorithmic Game Theory and Mechanism Design
Fall 2016

Allan Borodin (instructor)
Tyrone Strangway and Young Wu (TAs)

October 26, 2016

Lecture 13

- Announcements
  - Office hours: Tuesdays 3:30-4:30 SF 2303B
    or schedule meeting; or drop by.
  - I have posted the three questions for Assignment 2 which is due this Friday, October 28. This is now the entire assignment. Assignment must be submitted by the start of the class (i.e. tutorial).
  - Term test will take place on Friday, November 4. There was a typo in the course information sheet where it said that the term test was Friday, November 5. I also indicated that I would give three weeks notice but forgot to do that. I can delay the term test by a week or leave the date as November 4.
  - No class on Monday, November (fall break)
- Todays agenda
  - Continue discussion of Chapters 15 and perhaps 16
    - ★ The proof of VCG trutfulness
    - ★ The good and bad aspects of VCG
    - ★ Sponsored search; VCG vs GSP

The truthful VCG mechanism

THe Vickery Clarke Groves (VCG) mechanism is a deterministic mechanism that is truthful (ex-post incentive compatible) and individually rational. In Chapter 16, it is stated and proven to be truthful for the general case (i.e. not restricted to single parameter auctions) whenever we have a quasi-linear utiltiy function $u_i = v_i - p_i$.

The truthful VCG mechanism

THe Vickery Clarke Groves (VCG) mechanism is a deterministic mechanism that is truthful (ex-post incentive compatible) and individually rational. In Chapter 16, it is stated and proven to be truthful for the general case (i.e. not restricted to single parameter auctions) whenever we have a quasi-linear utiltiy function $u_i = v_i - p_i$.

The general proof of truthfulness is given in Theorem 16.2.6. The proof is remarkably simple given the generality of this result.

The truthful VCG mechanism

THe Vickery Clarke Groves (VCG) mechanism is a deterministic mechanism that is truthful (ex-post incentive compatible) and individually rational. In Chapter 16, it is stated and proven to be truthful for the general case (i.e. not restricted to single parameter auctions) whenever we have a quasi-linear utiltiy function $u_i = v_i - p_i$.

The general proof of truthfulness is given in Theorem 16.2.6. The proof is remarkably simple given the generality of this result.

As the name might suggest, the Vickery auction is VCG applied to the special case of a single item auction. It is also the special case of the Vickery auction for the mutli item case when there are k copies of the same item and the winners are the agents with the top k values and the price is the $k+1$-st value. (Do not just quote result for question 2 of assignment.)

The truthful VCG mechanism

THe Vickery Clarke Groves (VCG) mechanism is a deterministic mechanism that is truthful (ex-post incentive compatible) and individually rational. In Chapter 16, it is stated and proven to be truthful for the general case (i.e. not restricted to single parameter auctions) whenever we have a quasi-linear utiltiy function $u_i = v_i - p_i$.

The general proof of truthfulness is given in Theorem 16.2.6. The proof is remarkably simple given the generality of this result.

As the name might suggest, the Vickery auction is VCG applied to the special case of a single item auction. It is also the special case of the Vickery auction for the mutli item case when there are k copies of the same item and the winners are the agents with the top k values and the price is the $k+1$-st value. (Do not just quote result for question 2 of assignment.)

Since we are considering social welfare (and not revenue), we do not have to consider reserve prices but as we have mentioned, reserve prices do not cause a problem for truthfulness.

The VCG mechanism in a figure

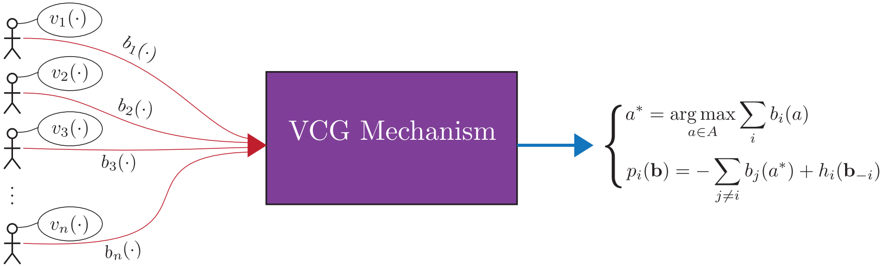


Figure: The depiction of VCG for arbitrary function $h_i(\mathbf{b}_{-i})$.

Here A is the set of feasible allocations. VCG is truthful for any h_i that does not depend on b_i. To make VCG IR and all payments non-negative, we use the "Clarke pivot":

$$h_i(\mathbf{b}_{-i}) = \max_{a \in A} \sum_{j \neq i} b_j(a)$$

VCG in words

VCG is a deterministic sealed bid auction. Each agent i submits their bids $b_i(a_i)$ for each desired outcome a_i.

Note: In the single-minded case, the agent has just one desired outcome but (say) in the general CA, an agent could desire many different sets each having its own valuation.

VCG in words

VCG is a deterministic sealed bid auction. Each agent i submits their bids $b_i(a_i)$ for each desired outcome a_i.

Note: In the single-minded case, the agent has just one desired outcome but (say) in the general CA, an agent could desire many different sets each having its own valuation.

The mechanism computes an optimal feasible allocation a^* with regard to social welfare. Each agent is allocated whatever is determined by a^*. For the CA problem, an agent receives a desired subset or does not get allocated any set. (Ties can be broken arbitrarily.)

VCG in words

VCG is a deterministic sealed bid auction. Each agent i submits their bids $b_i(a_i)$ for each desired outcome a_i.

Note: In the single-minded case, the agent has just one desired outcome but (say) in the general CA, an agent could desire many different sets each having its own valuation.

The mechanism computes an optimal feasible allocation a^* with regard to social welfare. Each agent is allocated whatever is determined by a^*. For the CA problem, an agent receives a desired subset or does not get allocated any set. (Ties can be broken arbitrarily.)

Given this optimal allocation a^*, the payment $p_i(\mathbf{b})$ for agent i is :

$$p_i(\mathbf{b}) = \max_{a \in A} \sum_{j \neq i} b_j(a) - \sum_{j \neq i} b_j(a^*)$$

That is, agent i is being charged his impact on the social welfare of the other agents.

VCG in words

VCG is a deterministic sealed bid auction. Each agent i submits their bids $b_i(a_i)$ for each desired outcome a_i.

Note: In the single-minded case, the agent has just one desired outcome but (say) in the general CA, an agent could desire many different sets each having its own valuation.

The mechanism computes an optimal feasible allocation a^* with regard to social welfare. Each agent is allocated whatever is determined by a^*. For the CA problem, an agent receives a desired subset or does not get allocated any set. (Ties can be broken arbitrarily.)

Given this optimal allocation a^*, the payment $p_i(\mathbf{b})$ for agent i is :

$$p_i(\mathbf{b}) = \max_{a \in A} \sum_{j \neq i} b_j(a) - \sum_{j \neq i} b_j(a^*)$$

That is, agent i is being charged his impact on the social welfare of the other agents. Be sure you understand this statement!

VCG is truthful

By definition, VCG is optimal with respect to social welfare if we assume eveyone bids truthfully. We now prove that VCG is truthful so that we can assume everyone does bid truthfully.

VCG is truthful

By definition, VCG is optimal with respect to social welfare if we assume eveyone bids truthfully. We now prove that VCG is truthful so that we can assume everyone does bid truthfully.

Proof of VCG truthfulness

$p_i(\mathbf{b}) = \max_{a \in A} \sum_{j \neq i} b_j(a) - \sum_{j \neq i} b_j(a^*) = C - \sum_{j \neq i} b_j(a^*)$
where we use C ($= h(\mathbf{b}_{-i})$ in the figure) to denote that the first term does not depend on the bid of agent i.
Then $u_i(\mathbf{b}|v_i) = v_i(a^*) - p_i(\mathbf{b}) = v_i(a^*) + \sum_{j \neq i} b_j(a^*) - C$
Reporting his true value, the outcome would be some allocation $a^{'} = argmax_a(v_i(a) + \sum_{j \neq i} b_j(a))$ which maximizes social welfare given $(v_i, \mathbf{b}_{-i})$ as the input to the optimal allocation algorithm.
Thus $u_i(v_i, \mathbf{b}_{-i}|v_i) = v_i(a^{'}) + \sum_{j \neq i} b_j(a^{'}) - C$
$\geq v_i(a^*) + \sum_{j \neq i} b_j(a^*) - C = u_i(\mathbf{b}|v_i)$

The difficiencies of VCG

There is an interesting article by Ausubel and Milgrom (2006) entitled "The Lovely by Lonely Vickrey Auction" describing VCG and its virtues but also some of the reasons why VCG is generally *not* used in practice. For some reasons why VCG is problematic, see Example 16.2.7 ; namely,

- The auctioneer may receive no payment.
- The auction is open to collusion

In addition,

- Agents may not understand VCG and then perhaps still want to strategize in their bidding.
- As we have mentioned before , VCG requires an *optimal allocation* in order to guarantee truthfulness. An approximation allocation algorithm coupled with the VCG payment rule may not be truthful.

Returning to single parameter problems: Chapter 15

Recall the meaning of a *single parameter mechanism design problem*.

Returning to single parameter problems: Chapter 15

Recall the meaning of a *single parameter mechanism design problem*.
We mean that each agent has the same value for any allocation in which it is a winner. Recall that $\mathcal{L} =$ the collection of all feasible sets of winners.

Returning to single parameter problems: Chapter 15

Recall the meaning of a *single parameter mechanism design problem*. We mean that each agent has the same value for any allocation in which it is a winner. Recall that $\mathcal{L}$ = the collection of all feasible sets of winners.

For single parameter problems (but still quasi linear utltilies), the VCG pricing has what may appear to be a perhaps more intuitive description.

Returning to single parameter problems: Chapter 15

Recall the meaning of a *single parameter mechanism design problem*. We mean that each agent has the same value for any allocation in which it is a winner. Recall that $\mathcal{L} =$ the collection of all feasible sets of winners.

For single parameter problems (but still quasi linear utltilies), the VCG pricing has what may appear to be a perhaps more intuitive description.

Consider an auction where we are trying to maximize the social welfare and each agent i is trying to maximize their utitility $u_i = v_i - p_i$. Once we have found an optimum solution (i.e. an optimum set of winners), we set the VCG prices where each winner pays his *threshold price*:

$$p_i = \max_{L \in \mathcal{L}_{\rangle}^{-}} b(L) - \max_{L \in \mathcal{L}_{\rangle}^{+}} b(L)$$

where $\mathcal{L}_i^- = \{L \in \mathcal{L} | i \notin L\}$ and $\mathcal{L}_i^+ = \{S | S \cup \{i\} \in \mathcal{L} | i \notin S\}$

Returning to single parameter problems: Chapter 15

Recall the meaning of a *single parameter mechanism design problem*. We mean that each agent has the same value for any allocation in which it is a winner. Recall that $\mathcal{L}$ = the collection of all feasible sets of winners.

For single parameter problems (but still quasi linear utltilies), the VCG pricing has what may appear to be a perhaps more intuitive description.

Consider an auction where we are trying to maximize the social welfare and each agent i is trying to maximize their utitility $u_i = v_i - p_i$. Once we have found an optimum solution (i.e. an optimum set of winners), we set the VCG prices where each winner pays his *threshold price*:

$$p_i = \max_{L \in \mathcal{L_i}^-} b(L) - \max_{L \in \mathcal{L_i}^+} b(L)$$

where $\mathcal{L}_i^- = \{L \in \mathcal{L} | i \notin L\}$ and $\mathcal{L}_i^+ = \{S | S \cup \{i\} \in \mathcal{L} | i \notin S\}$

This is not the same as the *critical price* that we will mention with regard to combinatorial auctions (CAs).

VCG and some single parameter examples in Chapter 15

shared communication channel.

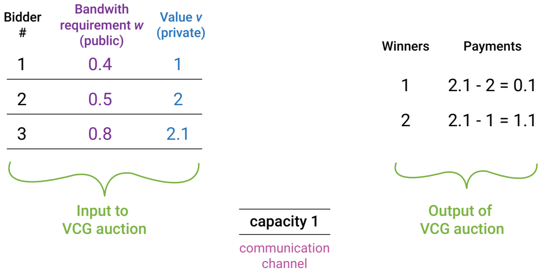


Figure: Figure 15.2: VCG for a small communication channel example

The optimal allocation is to allocate to agents 1 and 2. Without agent 1, the social welfare for the other agents is $v_3 = 2.1$ and the social welfare for the other agents with agent 1 is 2 so that agent 1 pays 2.1-2 $= 0.1$; similarly, agent 2 pays 2.1-1 $= 1.1$.

Some comments on the shared communication channel problem

As we noted, the underyling allocation problem is the knapsack problem which is a (weakly) NP-hard problem.

Some comments on the shared communication channel problem

As we noted, the underyling allocation problem is the knapsack problem which is a (weakly) NP-hard problem.

Assuming all values are fractional, we can scale all the inputs so that they are integers. If, after scaling, the capacity C, or all the weights $\{w_i\}$, or all the values $\{v_i\}$ are bounded by a polynomial (in n), then there is an optimal polynomial time algorithm.

Some comments on the shared communication channel problem

As we noted, the underyling allocation problem is the knapsack problem which is a (weakly) NP-hard problem.

Assuming all values are fractional, we can scale all the inputs so that they are integers. If, after scaling, the capacity C, or all the weights $\{w_i\}$, or all the values $\{v_i\}$ are bounded by a polynomial (in n), then there is an optimal polynomial time algorithm.

There is also an FPTAS algorithm for the knapsack problem and by a result of Briest, Krysta and Vöcking (2005), there is a monotone (in each v_i) FPTAS for this problem and this implies that this can be made into a truthful mechanism.

The s-CA example

An s-CA is a CA in which each agent only desires sets of size at most s. However, we assume "free disposal" in the sense that $v_i(S) \leq v_i(T)$ for any $S \subseteq T$.

The s-CA example

An s-CA is a CA in which each agent only desires sets of size at most s. However, we assume "free disposal" in the sense that $v_i(S) \leq v_i(T)$ for any $S \subseteq T$.

If each agent i only desires a single set S_i, this is called the "single-minded case".

The s-CA example

An s-CA is a CA in which each agent only desires sets of size at most s. However, we assume "free disposal" in the sense that $v_i(S) \leq v_i(T)$ for any $S \subseteq T$.

If each agent i only desires a single set S_i, this is called the "single-minded case".

The underlying allocation problem (i.e. set packing) is NP-hard for $s \geq 3$. However, there is are simple greedy algorithms that provide an s-approximation. Namely, one can either sort bids so that $b_1 \geq b_2 \ldots \geq b_n$ or so that $b_1/s_1 \geq b_2/s_2 \ldots \geq b_n/s_n$ where s_i is the size of the set S_i desired by agent i. The algorithm then just accepts each bid "greedily"; i.e. if the desired subset doesn't conflict with a previously accepted set.

The s-CA example

An s-CA is a CA in which each agent only desires sets of size at most s. However, we assume "free disposal" in the sense that $v_i(S) \leq v_i(T)$ for any $S \subseteq T$.

If each agent i only desires a single set S_i, this is called the "single-minded case".

The underlying allocation problem (i.e. set packing) is NP-hard for $s \geq 3$. However, there is are simple greedy algorithms that provide an s-approximation. Namely, one can either sort bids so that $b_1 \geq b_2 \ldots \geq b_n$ or so that $b_1/s_1 \geq b_2/s_2 \ldots \geq b_n/s_n$ where s_i is the size of the set S_i desired by agent i. The algorithm then just accepts each bid "greedily"; i.e. if the desired subset doesn't conflict with a previously accepted set.

This greedy algorithm is monotone (in each v_i) and monotone in each S_i in the sense that if (v_i, S_i) is a winner (given the other bids), then it remains a winner for (v_i', S_i') if $v_i' \geq v_i$ and/or if $S_i' \subseteq S_i$.

The good and bad new for s-CAS and CAs

For an arbitrary CA, these monotone greedy algorithms will (in the worst case) only achieve the "naive" approximation factor $\min(n, m)$.

The good and bad new for s-CAS and CAs

For an arbitrary CA, these monotone greedy algorithms will (in the worst case) only achieve the "naive" approximation factor $\min(n, m)$.

Somewhat surprisingly, Gonen and Lehman (2000) show that by sorting so that $b_1/\sqrt{s_1} \geq b_2/\sqrt{s_2} \ldots \geq b_n/\sqrt{s_n}$, and accepting greedily, the approximation factor is $\min(n, 2\sqrt{m})$.

The good and bad new for s-CAS and CAs

For an arbitrary CA, these monotone greedy algorithms will (in the worst case) only achieve the "naive" approximation factor $\min(n, m)$.

Somewhat surprisingly, Gonen and Lehman (2000) show that by sorting so that $b_1/\sqrt{s_1} \geq b_2/\sqrt{s_2} \ldots \geq b_n/\sqrt{s_n}$, and accepting greedily, the approximation factor is $\min(n, 2\sqrt{m})$.

Mu'alem and Nisan (2008) showed that in the single minded case, any such monotone greedy approximation algorithm can be made into a truthful mechanism (preserving the approximation factor) by using *critical prices* (i.e. smallest price for a winner to remain a winner when other bids are not changed) even if the both v_i and S_i are private information.

The good and bad new for s-CAS and CAs

For an arbitrary CA, these monotone greedy algorithms will (in the worst case) only achieve the "naive" approximation factor $\min(n, m)$.

Somewhat surprisingly, Gonen and Lehman (2000) show that by sorting so that $b_1/\sqrt{s_1} \geq b_2/\sqrt{s_2} \ldots \geq b_n/\sqrt{s_n}$, and accepting greedily, the approximation factor is $\min(n, 2\sqrt{m})$.

Mu'alem and Nisan (2008) showed that in the single minded case, any such monotone greedy approximation algorithm can be made into a truthful mechanism (preserving the approximation factor) by using *critical prices* (i.e. smallest price for a winner to remain a winner when other bids are not changed) even if the both v_i and S_i are private information.

In the multi minded case, with just 2 agents and $s \geq 3$, the same greedy algorithm becomes an $s + 1$ approximation. However, now this algorithm cannot be made into a truthful mechanism. In fact, for a wide class of greedy algorithms, Borodin and Lucier (2016) show that any greedy truthful mechanism (for this muiti minded case) can be at best an $\Omega(n, m)$ approximation for n agents and a universe of m items.

Sponsored search auctions

Section 15.5 presents a simplified but still insightful view of sponsored advertisements as occurs in a search engine.

Sponsored search auctions

Section 15.5 presents a simplified but still insightful view of sponsored advertisements as occurs in a search engine.

Here given a search request, agents (advertisers who think this search request will be by someone who might be interested in their product) bid (online) for one of say k sponsored slots.

Sponsored search auctions

Section 15.5 presents a simplified but still insightful view of sponsored advertisements as occurs in a search engine.

Here given a search request, agents (advertisers who think this search request will be by someone who might be interested in their product) bid (online) for one of say k sponsored slots.

It is assumed that slots are listed so that $c_1 \geq c_2 \ldots \geq c_k$ where c_j is the clickthrough rate of slot j. This means that someone making the given search request, will click on the jth slot with probability c_j.

Sponsored search auctions

Section 15.5 presents a simplified but still insightful view of sponsored advertisements as occurs in a search engine.

Here given a search request, agents (advertisers who think this search request will be by someone who might be interested in their product) bid (online) for one of say k sponsored slots.

It is assumed that slots are listed so that $c_1 \geq c_2 \ldots \geq c_k$ where c_j is the clickthrough rate of slot j. This means that someone making the given search request, will click on the jth slot with probability c_j.

If the ith bidder (advertiser) has value v_i for someone clicking on their advertisement, then the value for agent i obtaining slot j is $v_i \cdot c_j$.

Sponsored search auctions

Section 15.5 presents a simplified but still insightful view of sponsored advertisements as occurs in a search engine.

Here given a search request, agents (advertisers who think this search request will be by someone who might be interested in their product) bid (online) for one of say k sponsored slots.

It is assumed that slots are listed so that $c_1 \geq c_2 \ldots \geq c_k$ where c_j is the clickthrough rate of slot j. This means that someone making the given search request, will click on the jth slot with probability c_j.

If the ith bidder (advertiser) has value v_i for someone clicking on their advertisement, then the value for agent i obtaining slot j is $v_i \cdot c_j$.

Each advertiser will offer a bid of b_i per click. If the ith advertiser is successful in obtaining one of the k slots, she will pay a *price per click* (PPC) $p_i(\mathbf{b})$.

Sponsored search auctions

Section 15.5 presents a simplified but still insightful view of sponsored advertisements as occurs in a search engine.

Here given a search request, agents (advertisers who think this search request will be by someone who might be interested in their product) bid (online) for one of say k sponsored slots.

It is assumed that slots are listed so that $c_1 \geq c_2 \ldots \geq c_k$ where c_j is the clickthrough rate of slot j. This means that someone making the given search request, will click on the jth slot with probability c_j.

If the ith bidder (advertiser) has value v_i for someone clicking on their advertisement, then the value for agent i obtaining slot j is $v_i \cdot c_j$.

Each advertiser will offer a bid of b_i per click. If the ith advertiser is successful in obtaining one of the k slots, she will pay a *price per click* (PPC) $p_i(\mathbf{b})$. How should a search engine assign slots and set $p_i(\mathbf{b})$?

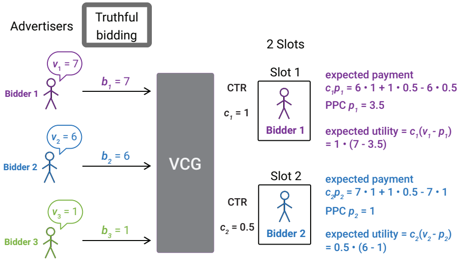
The VCG and GSP mechanisms for the sponsored search problem

Of course, we cannot be sure of what mechanism any search engine (e.g. Google) might use, but Facebook states that it uses VCG and Google states that it uses a mechanism called *Generalized Second Price*. Both mechanisms allocate the ith most valuable slot to the i^{th} highest bidder.

The VCG and GSP mechanisms for the sponsored search problem

Of course, we cannot be sure of what mechanism any search engine (e.g. Google) might use, but Facebook states that it uses VCG and Google states that it uses a mechanism called *Generalized Second Price*. Both mechanisms allocate the ith most valuable slot to the i^{th} highest bidder.

If bidders are truthful, then both VCG and GSP maximize social welfare. We know that VCG is a truthful mechanism so lets first consider VCG for the sponsored search problem.



VCG for sponsored search

Suppose we have k slots and we reorder bids (and bidders) so that $b_1 \geq b_2 \ldots \geq b_n$. The bidder $i \leq k$ gets slot i.

VCG for sponsored search

Suppose we have k slots and we reorder bids (and bidders) so that $b_1 \geq b_2 \ldots \geq b_n$. The bidder $i \leq k$ gets slot i.

Recall that slots are ordered so that $c_1 \geq c_2 \ldots c_k > c_{k+1} = 0$.

VCG for sponsored search

Suppose we have k slots and we reorder bids (and bidders) so that $b_1 \geq b_2 \ldots \geq b_n$. The bidder $i \leq k$ gets slot i.

Recall that slots are ordered so that $c_1 \geq c_2 \ldots c_k > c_{k+1} = 0$.

Consider the price for the advertiser winning the ith slot for $i \leq k$. The impact on the social welfare by the ith advertiser is to push advertisers $j > i$ down one slot. That is, he imposes an (expected) cost to the jth advertiser of $b_j(c_{j-1} - c_j)$ for a total impact of $\sum_{j=i+1}^{k+1} b_j(c_{j-1} - c_j)$.

Hence we want to charge the ith advertiser a per click bid $p_i(\mathbf{b})$ so that his expected cost is equal to the impact on the other advertisers; that is,

$$c_i p_i(\mathbf{b}) = \sum_{j=i+1}^{k+1} b_j(c_{j-1} - c_j) \quad \text{so that}$$

$$p_i(\mathbf{b}) = \sum_{j=i+1}^{k+1} b_j \frac{(c_{j-1} - c_j)}{c_i}$$

What if all slots had the same click through rate?

If all slots had the same click through rate then say
$c = c_1 = c_2 \ldots = c_k > c_{k+1} = 0$.

What if all slots had the same click through rate?

If all slots had the same click through rate then say $c = c_1 = c_2 \ldots = c_k > c_{k+1} = 0$.

Using VCG pricing, the per click pricing becomes

$$p_i(\mathbf{b}) = \sum_{j=i+1}^{k+1} b_j \frac{(c_{j-1} - c_j)}{c_i} = \sum_{j=i+1}^{k} b_j \frac{(c - c)}{c} + b_{k+1} \frac{c - 0}{c} = b_{k+1}$$

Does this look familiar?

What if all slots had the same click through rate?

If all slots had the same click through rate then say
$c = c_1 = c_2 \ldots = c_k > c_{k+1} = 0$.

Using VCG pricing, the per click pricing becomes

$$p_i(\mathbf{b}) = \sum_{j=i+1}^{k+1} b_j \frac{(c_{j-1} - c_j)}{c_i} = \sum_{j=i+1}^{k} b_j \frac{(c - c)}{c} + b_{k+1} \frac{c - 0}{c} = b_{k+1}$$

Does this look familiar?

When all slots have the same click through rate, this is just a multi-unit auction for a given item (i.e. selling k identical items).