Combinatorial Auctions

and LONGEST PATH for DAGs – A Case Study

Summer School on Computational Social Choice, San Sebastián 2016

Janosch Döcker[a]

[a]Joint work with Britta Dorn and Dominikus Krüger.

Combinatorial auctions

Definition 1 (WINNER DETERMINATION)**.** *Given:*

- *Goods* $A := \{1, 2, \dots, n\}$ *and*
- *bids on* combinations $C \subseteq A$. *A bid can be regarded as a pair of a subset of goods and a positive integer.*

Task: *Find revenue maximizing subset of bids, the winners, such that the corresponding combinations are pairwise disjoint.*

For each combination C we only represent the highest bid $b(C)$, which can be found by a simple and efficient preprocessing step [RPH98, p. 1136]. Since WINNER DETERMINATION is $\mathcal{NP}$-hard [RPH98] in general, a crucial aspect of practical applicability is finding efficiently solvable special cases.

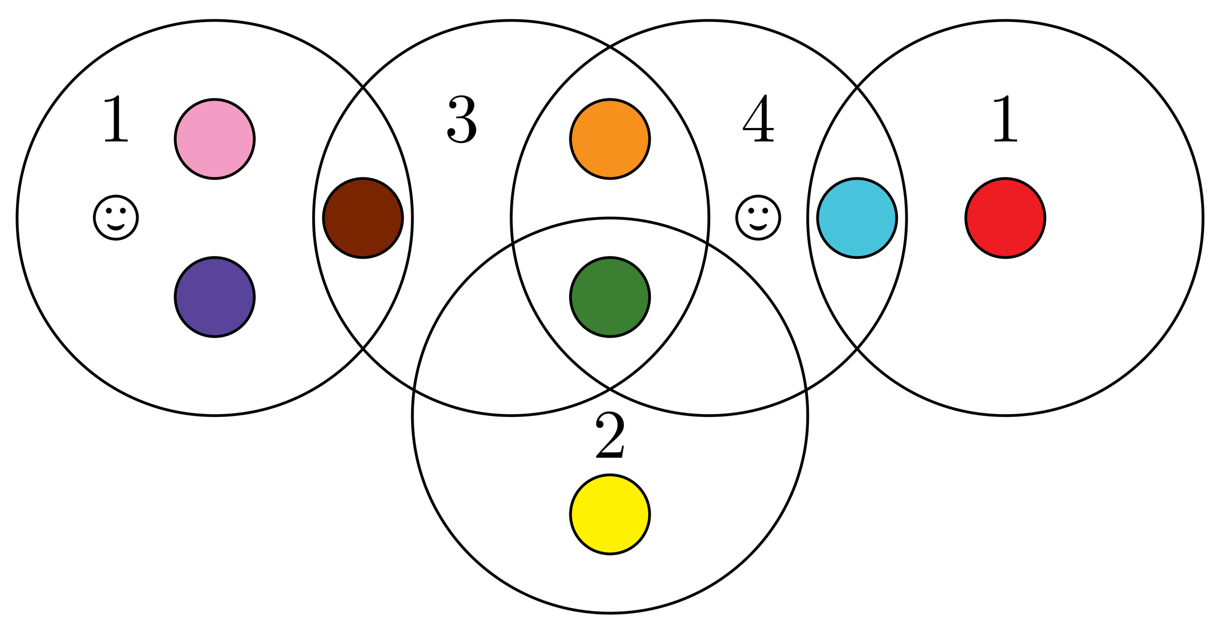

Figure 1: *Instance of* WINNER DETERMINATION. *The smileys mark the winners.*

Discrete intervals, LONGEST PATH

Rothkopf et al. [RPH98] identified several tractable instances of WINNER DETERMINATION, e. g., for bids on consecutive integers

$$[i, j] := \{x \in A : i \le x \le j\}, \quad i, j \in A,$$

we call such combinations *discrete intervals*. If bidders are allowed to place a bid on the union of two discrete intervals, WINNER DETERMINATION becomes $\mathcal{NP}$-hard again [CDS04, Theorem 4]. The union of two discrete intervals can also be regarded as one discrete interval with a gap, i.e., a combination of the form

$$[i, j] \setminus [x, y], \quad i < x \le y < j.$$

In the following we present a special case of this structure and show that it is tractable by reducing it to LONGEST PATH in a DAG.

Definition 2 (LONGEST PATH)**.** *Given a DAG* $G = (V, E)$ *with edge weights defined by a mapping* $g \colon E \to \mathbb{N}$ *and two vertices* $v_i, v_f \in V$, *find a directed path* $\pi = (v_i = v_1, v_2, \dots, v_f = v_{|\pi|})$ *that maximizes the path length* $\sum_{l=1}^{|\pi|-1} g((v_l, v_{l+1}))$.

LONGEST PATH in a DAG can be solved in time $\mathcal{O}(|V| + |E|)$ [SW11, p. 661].

Funnel structure

Definition 3 (Funnel)**.** *Let* $A := \{1, 2, \dots, n\}$ *be a set of goods. A set of combinations*

$$\mathcal{F} \subset \{[i, j] \setminus [x, y] : 1 \le i < x \le y < j \le n\}$$

is called a funnel, if there is an injective mapping $f \colon \mathcal{F} \to \{1, 2, \dots, |\mathcal{F}|\}$, *such that for all* $C, C' \in \mathcal{F}$ *with* $C \neq C'$ *the following holds:*

$$f(C) < f(C') \Rightarrow i \le i' \text{ and } j' \le j.$$

We can show that the number of combinations of any funnel $\mathcal{F}$ is bounded by $|\mathcal{F}| \le \frac{n^3}{6} - \frac{n^2}{2} + \frac{n}{3}$, where n is the number of goods; hence, we have $|\mathcal{F}| \in \mathcal{O}(n^3)$.

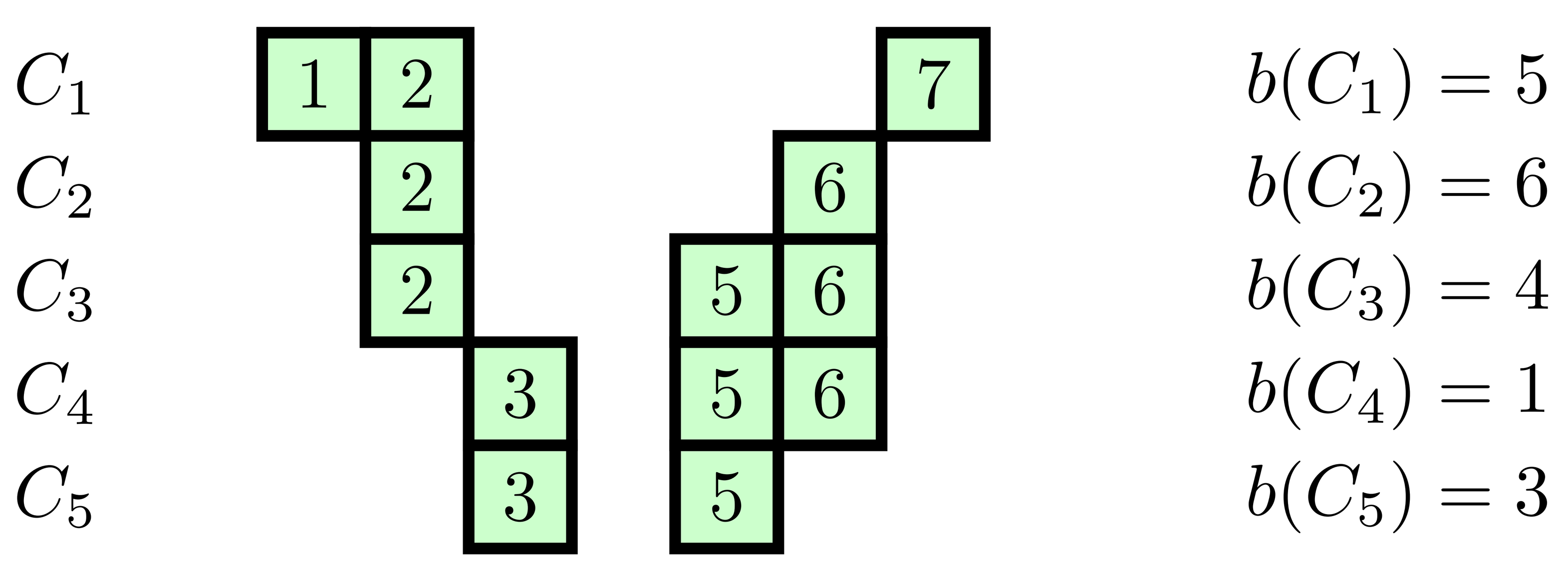

Figure 2: *A funnel* $\mathcal{F}$ *with corresponding bids.*

Reduction to LONGEST PATH in a DAG

- Given: a funnel $\mathcal{F}$ and bids represented by $b \colon \mathcal{F} \to \mathbb{N} \setminus \{0\}$.
- For each combination $[i, j] \setminus [x, y] \in \mathcal{F}$ we create two vertices $v_{i,j}$ and $v_{x,y}$ (if a vertex exists already, we do **not** introduce a copy), and the edge $(v_{i,j}, v_{x,y})$ with weight $b([i, j] \setminus [x, y])$.
- The intuition behind a vertex $v_{q,r}$ is that the goods in $[q, r]$ are available. In $v_i := \arg\max_{v_{i,j} \in V_e} (j - i)$ all goods of the funnel are available and in v_f none is available by definition, where v_f is a special new vertex.
- Since we do not require all goods to be sold, we have to ensure that from a vertex $v_{s,t}$ all $v_{i,j}$ with an outgoing weighted edge, $v_{i,j} \neq v_{s,t}$ and $s \le i \le j \le t$ are reachable. If there is no edge $(v_{s,t}, v_{i,j})$ with positive weight, we introduce this edge with weight 0.
- Finally, we connect each vertex $v_{x,y}$ without an outgoing edge directly to v_f, i.e., we introduce the edge $(v_{x,y}, v_f)$ with weight 0 (this happens if and only if no combination of the funnel is a subset of $[x, y]$).
- We call the resulting Graph $G_{\mathcal{F}}$. The longest path (with respect to edge weights) corresponds to an optimal solution of the given instance of WINNER DETERMINATION.

Figure 3 shows the result of this construction for the example shown in Figure 2.

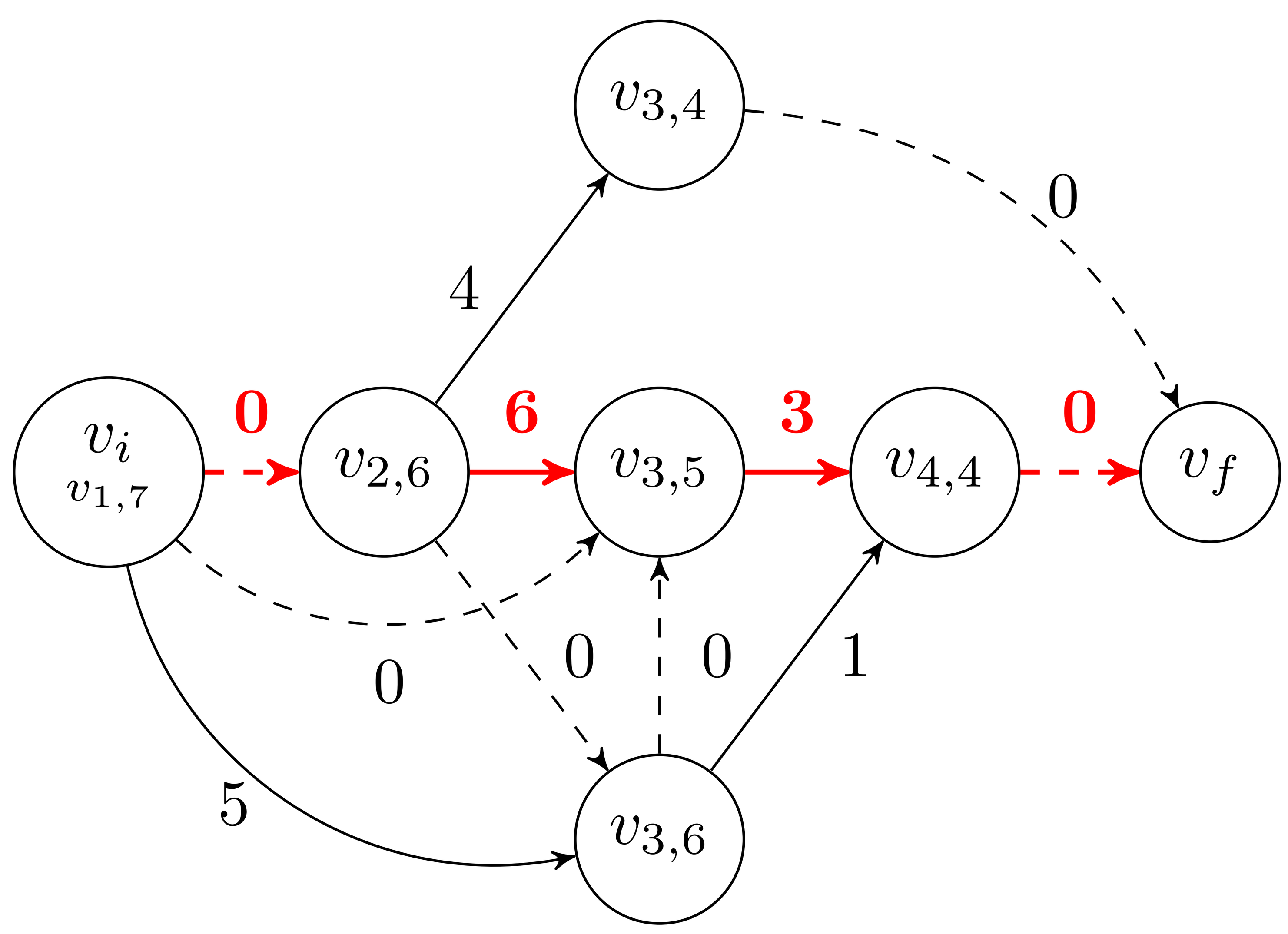

Figure 3: *Graph* $G_{\mathcal{F}}$ *for the funnel* $\mathcal{F}$ *and the bids shown in Figure 2. The dashed edges have weight 0. The longest path from* v_i *to* v_f *(marked in red) yields the winners.*

Results

By analyzing the construction described above, we can prove the following theorem.

Theorem. *Let* $A := \{1, 2, \dots, n\}$ *be a set of goods and* $\mathcal{F}$ *be a funnel. For these combinations* WINNER DETERMINATION *can be solved in time* $\mathcal{O}(n^3)$.

- Dynamic programming formulation with the same time complexity
- Extension of funnels to include intervals without gaps (the intervals must also satisfy the condition of Definition 3) $\rightsquigarrow$ same asymptotic complexity

References

[CDS04] Vincent Conitzer, Jonathan Derryberry, and Tuomas Sandholm. Combinatorial auctions with structured item graphs. In *AAAI*, 2004.

[RPH98] Michael H. Rothkopf, Aleksandar Pekeč, and Ronald M. Harstad. Computationally manageable combinational auctions. *Management Science*, 44(8):1131–1147, August 1998.

[SW11] Robert Sedgewick and Kevin Wayne. *Algorithms, 4th Edition*. Addison-Wesley Professional, 2011.

Universität Tübingen ·Wilhelm-Schickard-Institut für Informatik ·**Mathematische Strukturen in der Informatik**
Sand 13 · 72076/Tübingen · Germany · Telefon +49 7071 29-70499 · janosch.doecker@uni-tuebingen.de