STOCHASTIC LOCAL SEARCH
FOUNDATIONS AND APPLICATIONS

# Combinatorial Auctions

Holger H. Hoos & Thomas Stützle

# Introduction

Auctions are attractive as mechanisms for ...

- E-commerce (business transactions, sales)
- resource allocation
  (radio frequencies, pollution rights, scheduling)
- coordinating agents in multi-agent systems

$\leadsto$ *AI community shows an increasing interest in auctions*

**Standard, single item auctions:**

- winner determination is easy
- can't express complementarities of goods

*Example:*

Auctioning flight tickets: people want specific combination of travel dates.

Bidder A wants to fly from Austin to London over the weekend, thus has to bid for Fri (or Sat) and Sun flights.

$\leadsto$ *bidding for bundles (= combinations of items)*

**Bidding for bundles ...**

- minimises buyer's risk of getting incomplete combinations
- allows to express preferences more directly

Bids for combinations of items can overlap
$\rightsquigarrow$ winner determination is hard!

# Combinatorial Auctions

**Given:**

- set of goods $G = \{g_1, \ldots, g_n\}$ to be auctioned
- set of bids $B = \{(c_i, v_i) | c_i \subseteq G, v_i \in R^+\}$

**Definition:**

An *assignment* is a function $A : G \mapsto B$ mapping goods to bids.

A bid $b_i = (c_i, v_i)$ is *satisfied by* $A$ if $\forall g \in c_i : A(g) = b_i$.

**Winner determination problem for CAs:**

Find an assignment $A$ which maximises the total revenue, i.e., $\sum\{v_i | b_i = (c_i, v_i) \in B \wedge b_i \text{satisfied by} A\}$.

**Bad news:** This problem (decision version) is NP-complete.

**Even worse:** It is not even polynomially approximable within $k \leq n^{1-\epsilon}$.

**Schematic Bids**

Standard CAs can't express *substitutabilities* (or indifference) between goods.

*Example (flight tickets):*

I want to make the weekend trip to London, but don't care whether I depart on Fri or Sat, and I am indifferent between returning on Sun or Mon.

$\leadsto$ can be expressed in standard CAs, but creates lots of bids

**Richer, logical languages for bids:**

- CNF bids: allow bids for bundles of disjunctions, e.g. (Fri $\vee$ Sat) $\wedge$ (Sun $\vee$ Mon)
- $k$-of bids: allow bids for bundles of subset selections, e.g. 2-of{Fri,Sat,Sun,Mon}$\wedge$ 2-of{Mon,Tue,Thu}

Considered here for the first time — no algorithms available!

# SLS Algorithms for CAs

**Stochastic local search is ...**

- popular and very successful for many combinbatorial problems (TSP, CSP, SAT, scheduling, planning, ...)
- typically easy to implement
- often easy to parallelise
- often the method of choice for hard, large problems

**CASLS family of SLS algorithms for CA:**

- *search space:* space of feasible, partial assignments
- *solution set:* assignments with optimal / given revenue
- *neighbourhood relation:* assignments reachable by reassigning set of goods such that one formerly unsat bid becomes satisfied
- *search initialisation function:* start with empty assignment
- *search step function:* picks a neighbour according to stochastic heuristic $h$

**Casanova algorithm:**

- member of the CASLS family using a heuristic which is closely related to one of the most successful SLS algorithms for SAT, Novelty$^+$ (Hoos 1999; McAllester,Selman,Kautz 1997).
- several parameters which can be optimised, including two noise parameters controlling the degree of randomness in $h$.
- conceptually very simple (compared to systematic algorithms)
- easy to implement

# Empirical Results

Tested Casanova against CASS, the better of the two known systematic algorithms for CAs.

Used a range of synthetic problem sets with various properties.

For each problem instance, measured

- mean solution quality (revenue) for fixed run-time
- mean run-time for finding optimal solutions

Compare these over the test-sets.

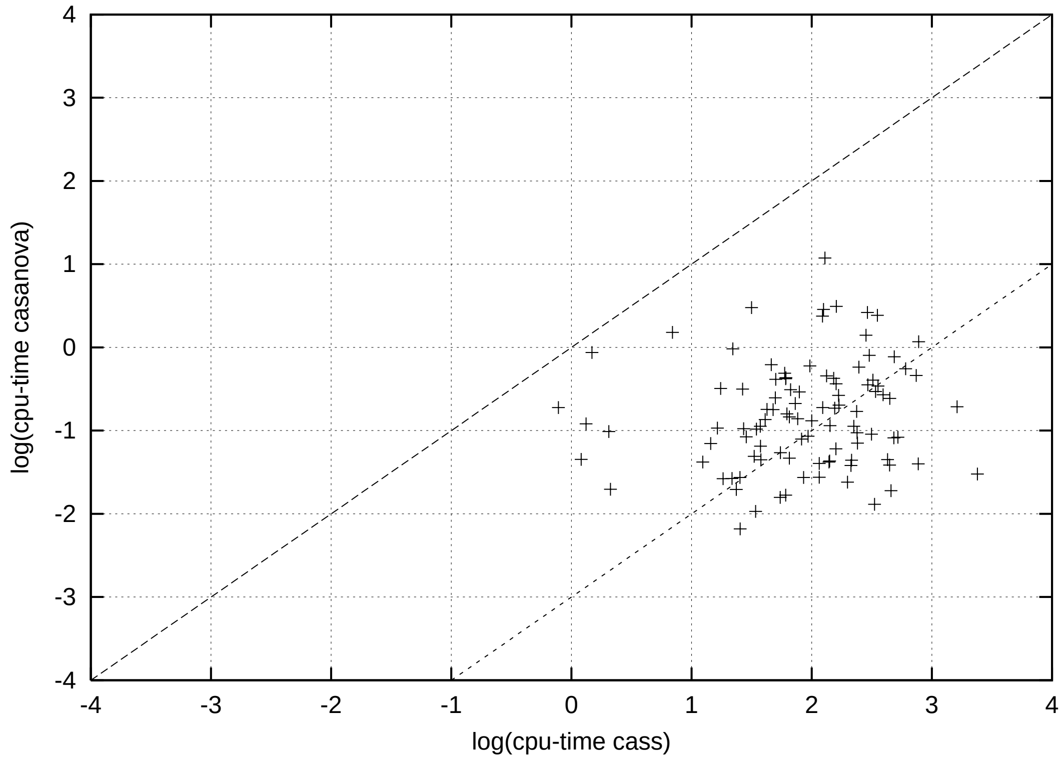
## Regular bids: Revenue for CASS vs. Casanova (fixed run-time)

| test-set | # inst | cutoff | CASS | | | Casanova | | |
|---|---|---|---|---|---|---|---|---|
| | | | median | $Q_{90}$ | $Q_{10}/Q_{90}$ | median | $Q_{90}$ | $Q_{10}/Q_{90}$ |
| UNI-3-100-1000 | 100 | 10s | 130396 | 133838 | 1.0489 | **134216** | 136203 | 1.0314 |
| UNI-3-200-2000 | 100 | 10s | 252084 | 257643 | 1.0438 | **264814** | 267573 | 1.0212 |
| UNI-3-100-5000 | 100 | 30s | 142947 | 144015 | 1.0172 | **143886** | 144666 | 1.0104 |
| UNI-3-200-10000 | 100 | 60s | 281413 | 284033 | 1.0189 | **286164** | 287632 | 1.0102 |
| BIN-0.01-500-5000 | 10 | 60s | 583279 | 594931 | 1.0367 | **616708** | 623624 | 1.0351 |
| DEC-0.75-500-5000 | 10 | 60s | 668458 | 678830 | 1.0380 | **675198** | 279919 | 1.0090 |
| EXP-5-100-1000 | 10 | 30s | **135027** | 135658 | 1.0298 | 132705 | 134412 | 1.0295 |
| EXP-5-500-5000 | 10 | 60s | 647629 | 650302 | 1.0240 | **655329** | 659238 | 1.0199 |

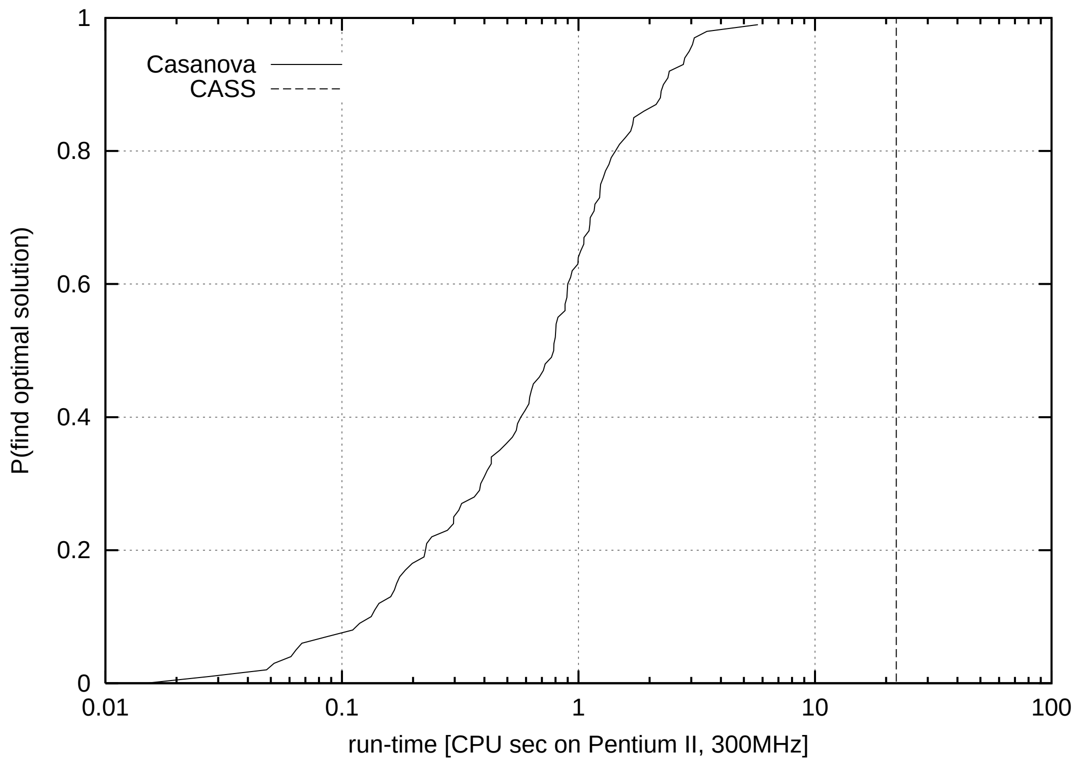
# Regular bids: Search cost for CASS vs. Casanova (finding optimal solutions)

| test-set | # inst | CASS | | | Casanova | | |
|---|---|---|---|---|---|---|---|
| | | median | $Q_{90}$ | $Q_{10}/Q_{90}$ | median | $Q_{90}$ | $Q_{10}/Q_{90}$ |
| UNI-3-50-50 | 100 | 0.058 | 0.125 | 9.09 | **0.0092** | 0.029 | 7.61 |
| UNI-3-75-75 | 100 | 2.211 | 6.222 | 10.91 | **0.030** | 0.197 | 24.99 |
| UNI-3-100-100 | 100 | 96.41 | 446.50 | 27.17 | **0.136** | 0.964 | 36.24 |
| UNI-3-50-100 | 100 | 0.487 | 1.40 | 15.20 | **0.091** | 0.543 | 21.29 |
| UNI-3-75-150 | 100 | 125.76 | 409.95 | 17.24 | **1.078** | 3.974 | 25.59 |
| UNI-3-20-2000 | 100 | 33.99 | 140.55 | 462.86 | **1.725** | 5.160 | 6.911 |
| UNI-10-200-200 | 100 | 147.99 | 308.92 | 15.72 | **1.677** | 6.051 | 10.54 |
| BIN-0.2-20-500 | 100 | **0.051** | 0.066 | 1.48 | 7.980 | 31.447 | 11.08 |
| DEC-0.75-200-200 | 10 | 252.82 | 1061.04 | 44.62 | **6.236** | 632.35 | 800.747 |
| EXP-5-20-500 | 100 | **0.0282** | 0.0315 | 1.21 | 0.852 | 8.689 | 749.01 |

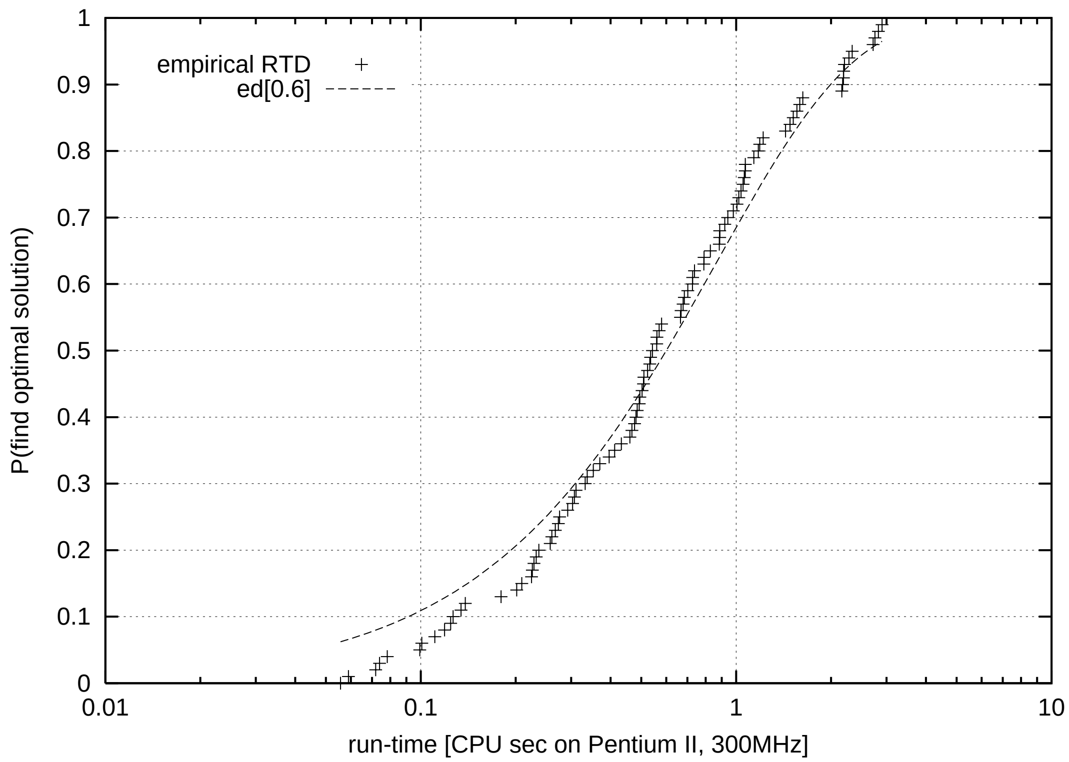
Correlation of search cost (finding optimal solution), test-set `UNI-3-100-100` (100 goods, 100 bids).



# Run-time distributions for finding optimal solution for a `UNI-3-100-100` (100 goods, 100 bids) instance



Run-time distributions for finding optimal solution for a `DEC-0.75-200-200` (200 goods, 200 bids) instance



# CNF and *k-of* bids: Revenue for CASS vs. Casanova (fixed run-time)

| test-set | # inst | cutoff | CASS | | | Casanova | | |
|---|---|---|---|---|---|---|---|---|
| | | | median | $Q_{90}$ | $Q_{10}/Q_{90}$ | median | $Q_{90}$ | $Q_{10}/Q_{90}$ |
| CUNI-3-50-50 | 100 | 10s | 55015 | 58479 | 1.1458 | **63360** | 65745 | 1.0900 |
| CUNI-3-100-100 | 100 | 60s | 104868 | 108687 | 1.1017 | **127011** | 130440 | 1.0613 |
| CUNI-3-50-250 | 100 | 60s | 52245 | 56943 | 1.1975 | **70158** | 70551 | 1.0176 |
| CPOIS-2-50-50 | 100 | 10s | 53204 | 56397 | 1.1542 | **60398** | 63115 | 1.1049 |
| CPOIS-2-100-100 | 100 | 60s | 99238 | 105275 | 1.1334 | **117889** | 122673 | 1.0691 |
| CPOIS-2-50-250 | 100 | 60s | 53066 | 56094 | 1.1272 | **69608** | 70755 | 1.0317 |
| CPOIS-2-100-500 | 100 | 60s | 101568 | 105941 | 1.1043 | **135973** | 138266 | 1.0323 |
| KUNI-2-4-2-100-100 | 10 | 60s | 48812 | 50608 | 1.1988 | **59938** | 63194 | 1.0940 |

## CNF and *k-of* bids: Search cost for CASS vs. Casanova (finding optimal solutions)

| test-set | # inst | CASS | | | Casanova | | |
|---|---|---|---|---|---|---|---|
| | | median | $Q_{90}$ | $Q_{10}/Q_{90}$ | median | $Q_{90}$ | $Q_{10}/Q_{90}$ |
| CUNI-3-20-20 | 100 | 791.11 | 2904.89 | 180.58 | **0.050** | 0.138 | 5.24 |
| CPOIS-2-20-20 | 100 | 1.855 | 9.355 | 41.15 | **0.240** | 1.048 | 17.74 |
| KUNI-2-4-2-20-20 | 100 | 24.364 | 48.690 | 72.40 | **0.474** | 4.678 | 24.40 |

**Summary of results:**

- Casanova outperforms CASS on large problems with fixed cutoff times
- Although incomplete, Casanova finds optimal solutions to small instances, often much faster than CASS
- Casanova's performance improves relative to CASS's with growing problem size
- Casanova can be easily parallelised with optimal speedup

$\rightsquigarrow$ *for solving large problem instances (*$>100$ *goods,* $>1000$ *bids), SLS algorithms like Casanova significantly better than current systematic search procedures*