On Greedy Algorithms and Approximate Matroids

a riff on Paul Milgrom’s “Prices and Auctions in Markets with Complex Constraints”

Tim Roughgarden (Stanford University)
A 50-Year-Old Puzzle

Persistent mystery: why do so many heuristics for optimization problems work so well in practice?
A 50-Year-Old Puzzle

Persistent mystery: why do so many heuristics for optimization problems work so well in practice?

Why should economists care?: real-world market design often requires heuristics.
A 50-Year-Old Puzzle

Persistent mystery: why do so many heuristics for optimization problems work so well in practice?

Why should economists care?: real-world market design often requires heuristics.

1. Computational reasons. Not enough time/computational power to solve exactly. [Nisan/Ronen 99, Lehmann/O’Callaghan/Shoham 99]

2. Economic reasons. Descending clock implementations require “reverse greedy” algorithms. [Milgrom/Segal 14]
Two Example Problems

**LOS99 setup:** each bidder has a desired bundle of items and a valuation for it.

- **goal:** select winners to max social welfare
  - subject to no item being allocated twice
Two Example Problems

**LOS99 setup:** each bidder has a desired bundle of items and a valuation for it.
- **goal:** select winners to max social welfare
  - subject to no item being allocated twice

**Milgrom-Segal/FCC Incentive Auction:** each station has valuation for its current license.
- **goal:** select subset of stations to max welfare
  - winners keep their licenses
  - subject to repacking winners into target # of channels
General Formalism

**Packing problem:** ground set $X$, collection $C$ of subsets ($C$ satisfies “free disposal/downward-closure”).
- each $x$ in $X$ has a nonnegative value $v_x$
- goal: choose $S$ from $C$ to maximize $\Sigma_{x \in S} v_x$

**Examples:** (all NP-hard)
- knapsack
- single-minded bidders (LOS99) [$\approx$ independent set]
- station repacking (MS14) [$\approx$ graph coloring]
Matroids: A Solvable Special Case

Definition: (X,C) is a matroid if .... [omitted]

One property: all maximal subsets of C have the same cardinality (the rank of the matroid).
Matroids: A Solvable Special Case

Definition: \((X, C)\) is a matroid if .... [omitted]

One property: all maximal subsets of \(C\) have the same cardinality (the rank of the matroid).

Example: acyclic subgraphs   Non-example: matchings
Matroids: A Solvable Special Case

**Definition:** 
\((X,C)\) is a *matroid* if .... [omitted]

**One property:** all maximal subsets of \(C\) have the same cardinality (the *rank* of the matroid).

**Example:** acyclic subgraphs  
**Non-example:** matchings

![acyclic subgraph](image1)

![matching](image2)

**Fact:** greedy algorithm always optimal iff matroid.
Approximate Matroids/Substitutes?

Substitutability index: [Milgrom]

$$\rho(C) := \max_{\text{matroid } R \subseteq C} \left( \min_{X \in C} \left( \max_{X' \subseteq X, X' \in R} \frac{|X'|}{|X|} \right) \right)$$
Approximate Matroids/Substitutes?

Substitutability index: [Milgrom]

\[
\rho(C) := \max_{\text{matroid } R \subseteq C} \left( \min_{X \in C} \left( \max_{X' \subseteq X} \frac{|X'|}{|X|} \right) \right)
\]

Heuristic #1: Define \( R^* = \arg\max \) above. Optimize over \( R^* \) (e.g., using greedy) instead of over \( C \).

Theorem: [Milgrom] for every \( C \), worst-case (over \( v_x \)'s) approximation is exactly \( \rho(C) \).
On the Substitutability Index

Open question: is the substitutability index close to 1 in the FCC Incentive Auction?
On the Substitutability Index

Open question: is the substitutability index close to 1 in the FCC Incentive Auction?

In general: substitutability index can be unreasonably small.

substitutability index of matchings in $K_{n,n}$?
On the Substitutability Index

Open question: is the substitutability index close to 1 in the FCC Incentive Auction?

In general: substitutability index can be unreasonably small.

substitutability index of matchings in $K_{n,n} = 1/n$
An Alternative Parameterization

Rank quotient: [Korte/Hausmann 78]

\[ \alpha(C) := \min_{X, X' \text{ maximal in } C} \frac{|X'|}{|X|} \]

Heuristic #2: Run greedy algorithm w.r.t. C.
- one pass through elements from highest to lowest
- add current element iff preserves feasibility
An Alternative Parameterization

Rank quotient: [Korte/Hausmann 78]

\[ \alpha(C) := \min_{S \subseteq X} \min_{X, X' \text{maximal in } \pi_S(C)} \frac{|X'|}{|X|} \]

Heuristic #2: Run greedy algorithm w.r.t. C.
- one pass through elements from highest to lowest
- add current element iff preserves feasibility
An Alternative Parameterization

Rank quotient: [Korte/Hausmann 78]

$$\alpha(C) := \min_{S \subseteq X} \min_{X,X' \text{ maximal in } \pi_S(C)} \frac{|X'|}{|X|}$$

Heuristic #2: Run greedy algorithm w.r.t. C.

- one pass through elements from highest to lowest
- add current element iff preserves feasibility

Theorem: [Korte/Hausmann] for every C, worst-case (over \(v_x\)’s) approximation is exactly \(\alpha(C)\).