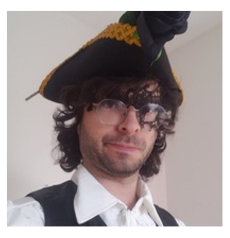



Paula Navarrete

Cyrus Cousins

Efficient Algorithms for Fair Course Allocation


Yair Zick

George Bissias

Fair & Explainable Decision-Making (FED) Lab

UMassAmherst | College of Information & Computer Sciences

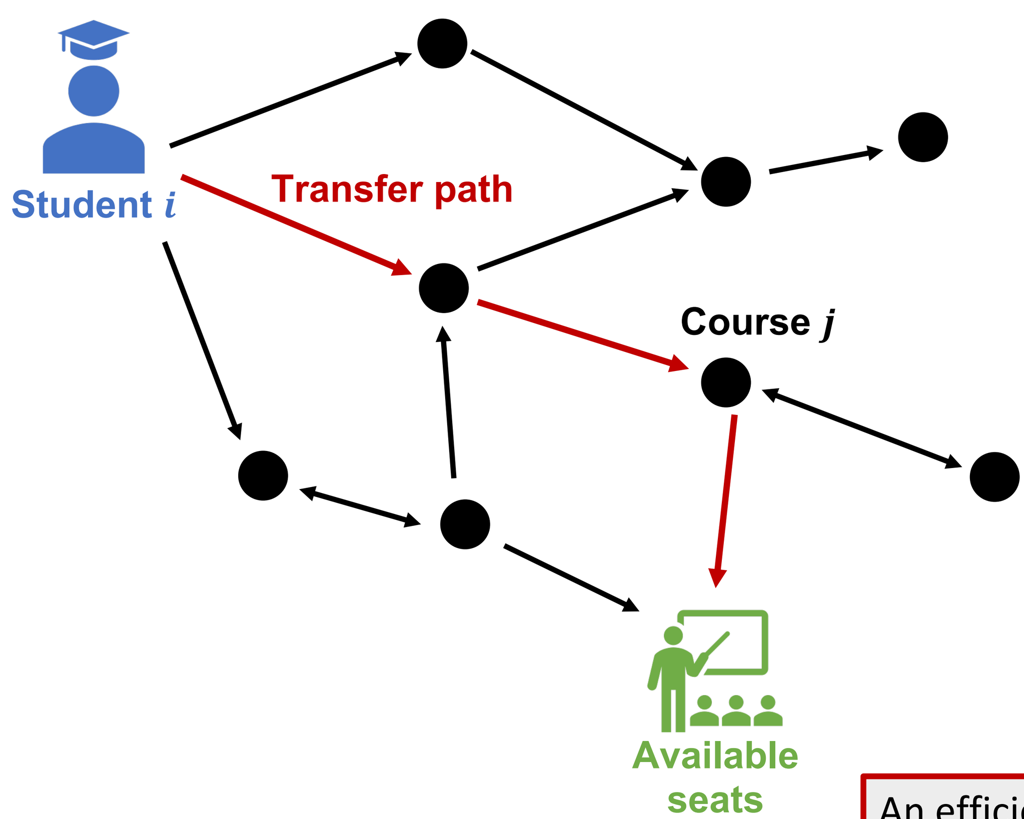
Yankee Swap with Duplicate Items

ALGORITHM : Yankee Swap

Input : Set of students N, set of classes M, and valuation functions $\{v_i\}_{i\in N}$
Output : A clean allocation X
Initialize exchange graph G
$X = (X_0, X, \dots, X_n) \leftarrow (M, \emptyset, \dots, \emptyset)$ // All seats initially in X_0
$U \leftarrow N$
while $U \neq \emptyset$ **do**
  $i \leftarrow \arg\max -v_k(X_k)$ // Pick lowest utility student
  find shortest path from student i to class $j \in X_0$
  if a path exists **then**
    update X
    $X_0[j] \leftarrow X_0[j] - 1$ // Reduce class seat by 1
    generate / **update** G
  else
    $U \leftarrow U \setminus i$ // Remove student
  end
end

Our algorithm considers updating the exchange graph rather than recomputing it from scratch in every iteration, leading to a significant reduction in running time!!!

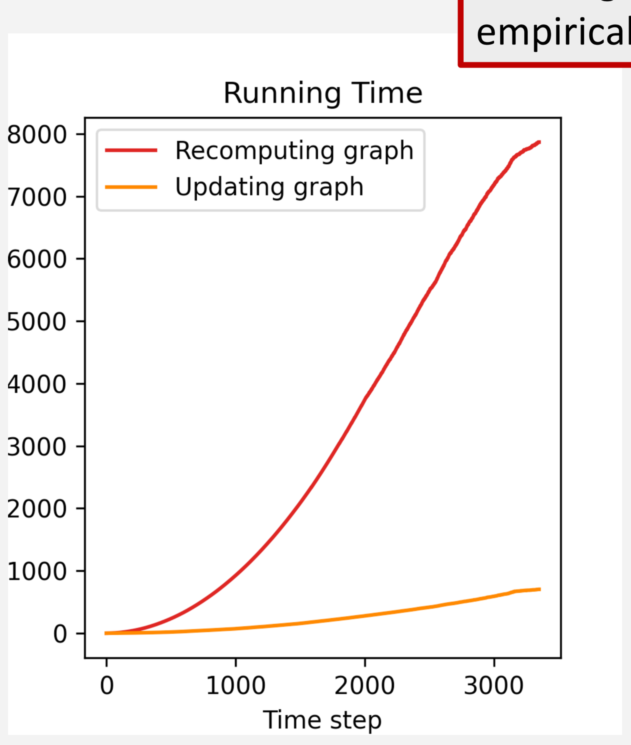
We implemented the Yankee Swap allocation algorithm considering students with binary submodular valuation functions (Viswanathan and Zick , 2023a), and incorporated multiplicity of items.



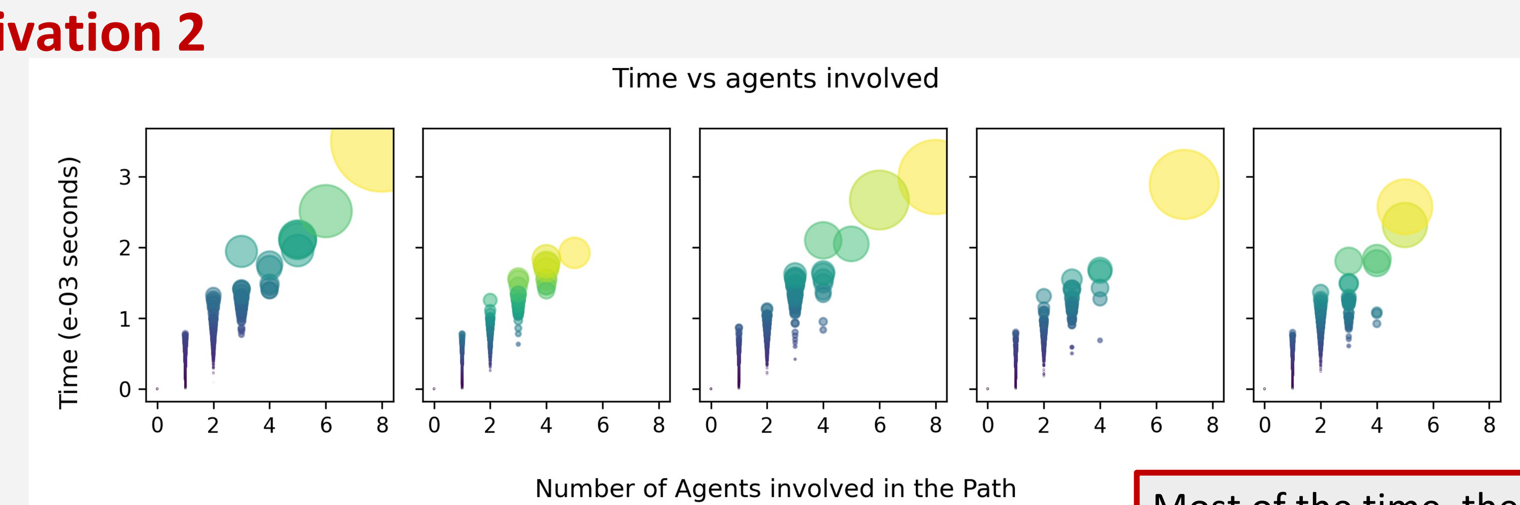
Runtime Analysis

Motivation 1

Updating instead of recomputing the exchange graph is empirically faster!



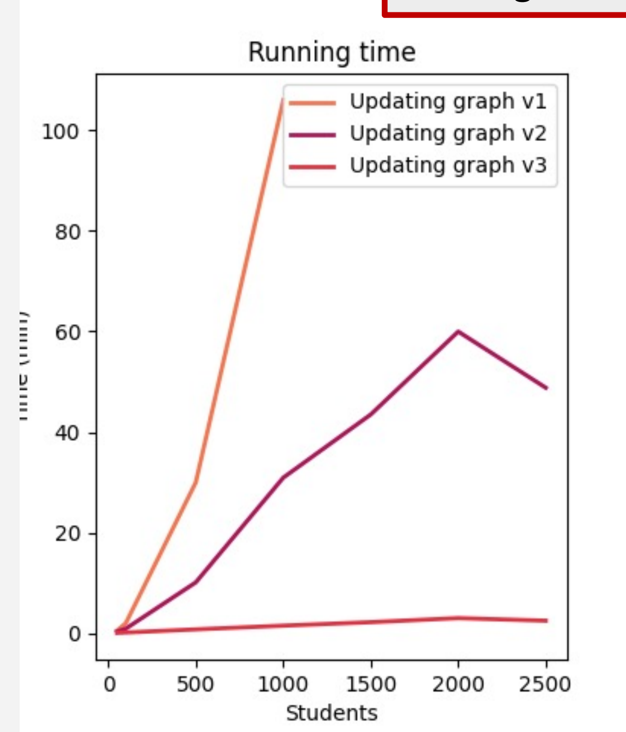
Motivation 2



Most of the time, the algorithm spends on updating the exchange graph. Can we do better?

Final version:

An efficient bookkeeping can considerably reduce running time!



Experiments

We ran YS with $m = 107$ courses from the UMass CS schedule and $n = 3000$ randomly generated students. We compare YS against three benchmark algorithms in terms of 5 different metrics.

Benchmark Algorithms:
Integer Linear Programming (**ILP**)
Round Robin (**RR**)
SPIRE Algorithm (**SPIRE**)

Performance Metrics

Let N be the set of students, and X an allocation of the items, where X_i is the bundle allocated to student i.
We are interested in

Maximizing

$$USW(X) = \frac{1}{|N|}\sum_{i\in N} v_i(X_i)$$

$$NSW(X) = \left(\prod_{i\in N_{>0}(X)} v_i(X_i)\right)^{\frac{1}{|N_{>0}(X)|}}$$

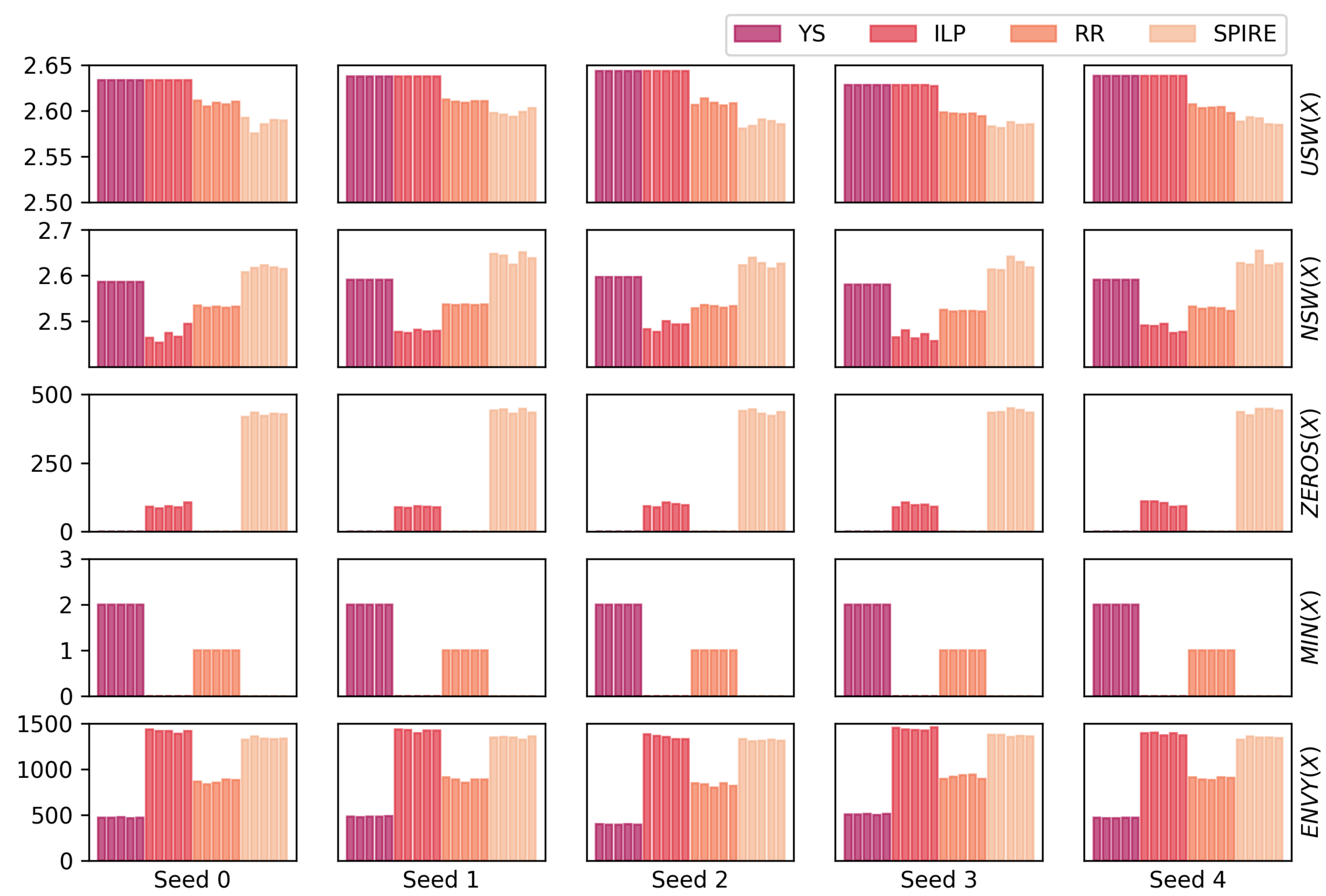
$$MIN(X) = \min_i v_i(X_i)$$

Minimizing

$$ZEROS(X) = |N \setminus N_{>0}(X)|$$

$$ENVY(X) = \left|\{i \in N | \exists j \in N : v_i(X_i) < v_i(X_j)\}\right|$$

Where $N_{>0}(X) = \{i \in N | v_i(X_i) > 0\}$



References
- Viswanathan, V., & Zick, Y. (2023a). Yankee swap: a fast and simple fair allocation mechanism for matroid rank valuations. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multi-Agent Systems* (AAMAS).
- Viswanathan, V., & Zick, Y. (2023b). A general framework for fair allocation under matroid rank valuations. In *Proceedings of the 24th ACM Conference on Economics and Computation* (pp. 1129-1152).

pnavarretedi@umass.edu