Efficient Yankee Swap for Fairly Allocating Courses to Students

Paula Navarrete
Cyrus Cousins
Yair Zick
Vignesh Viswanathan

Fair & Explainable Decision-Making (FED) Lab

UMassAmherst | College of Information & Computer Sciences

Yankee Swap with Duplicate Items

We implemented the Yankee Swap allocation algorithm considering students with binary submodular valuation functions (Viswanathan and Zick , 2023a), and incorporated duplicity of 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
$X = (X_0, X, \ldots, X_n) \leftarrow (M, \emptyset, \ldots, \emptyset)$ // 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
    $U \leftarrow N \setminus i$ // 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.

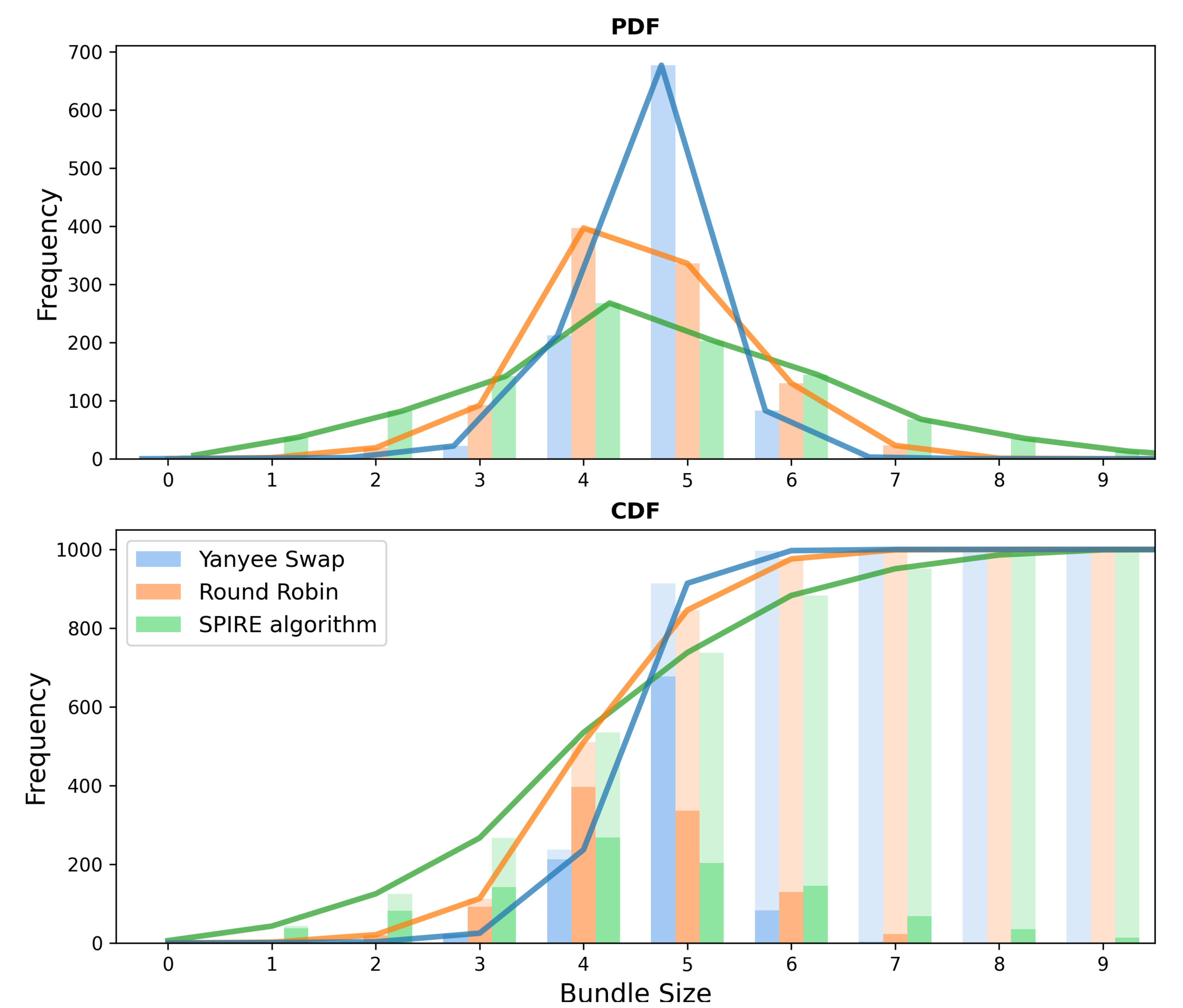
Vanilla Yankee Swap

We run Yankee Swap (YS) with $n = 107$ courses from the UMass CS schedule, and $m = 1000$ randomly generated students. We compare YS against the current course allocator (SPIRE) and Round Robin (RR).

	YS	RR	SPIRE
USW*	4.9	4.6	4.5
Nash Zeros*	0	0.2	4.4
NSW*	4.9	4.5	4.2
EF	0	16.4	1985
EF-X*	0	0	197
EF-1	0	16	1975

* YS is guaranteed to maximize USW and NSW, and is also envy free up to any item (EF-X)

Results show average values for output allocations obtained for 5 different seeds.

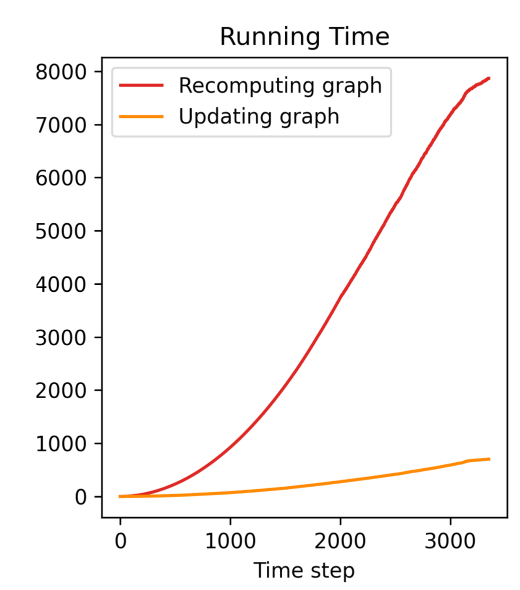


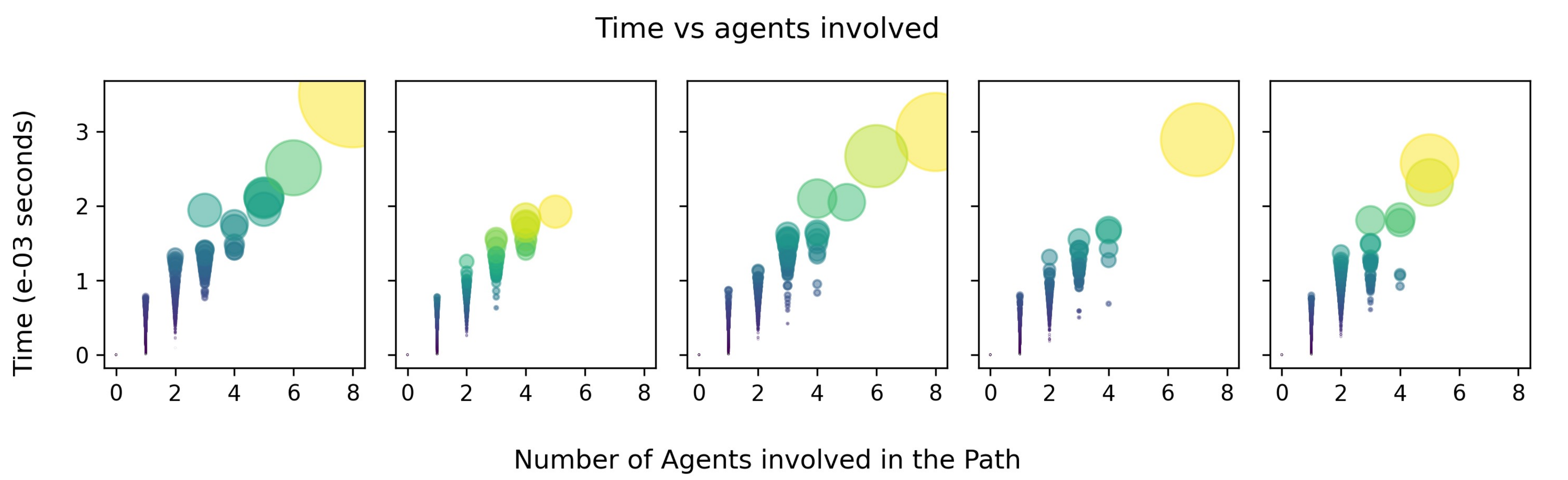
Runtime Analysis

	YS
Individual Items	$O(S^2(n+\tau)(n+S))$
Recomputing graph	$O(m(n+S)(n+mS_{MAX}\tau))$
Updating graph	$O((n+S)(\ln n + m^2 + p\gamma C_{MAX}(S_{MAX}+\tau))$

τ: maximum time to compute $v_i(X_i)$
C_{MAX}: student class limit
γ: Max number of desired classes
$S = \sum_{j \in M} S_j$, S_j are the seats in class j
$S_{MAX} = \max_j S_j$
$p =$ length of the path

Updating instead of recomputing the exchange graph is empirically faster!

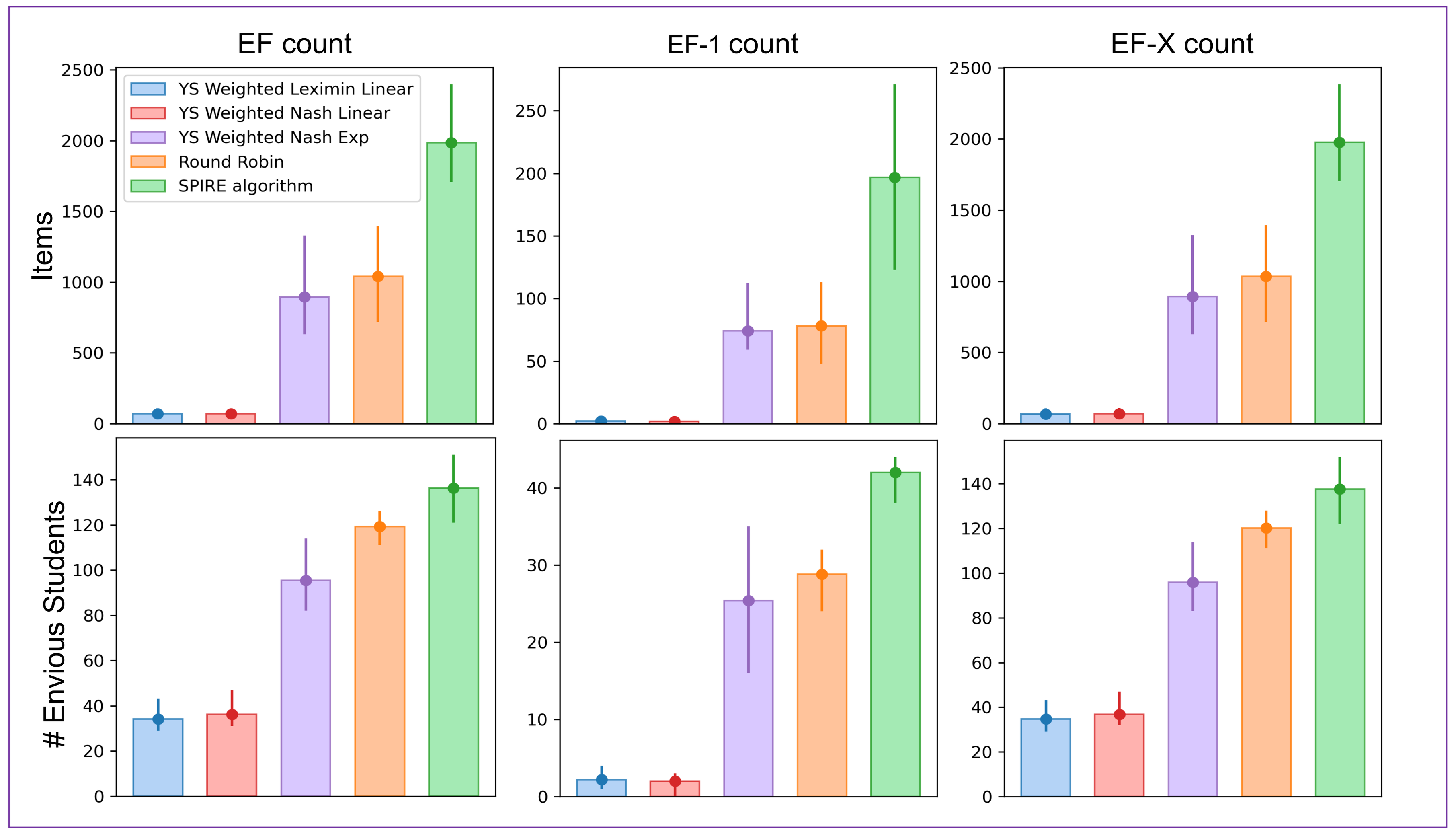




Motivation: Most of the time, the algorithm spends on updating the exchange graph, can we do better?

General Yankee Swap

What happens if we consider students with different weights w_i? General Yankee Swap is a generalized version of the algorithm that allows maximizing any justice criteria (Viswanathan and Zick, 2023b). Vanilla Yankee Swap is guaranteed to be EF-X. What is the performance in terms of enviousness for the general version?



Generalized YS **only alters the order in which we pick the students**. Instead of maximizing $-v_i(X_i)$, we might maximize:

Weighted Leximin:
$$-\frac{v_i(X_i)}{w_i}$$

Weighted Nash:
$$\left(1+\frac{1}{v_i(X_i)}\right)^{w_i} \text{ if } v_i(X_i) > 0,$$
a large C otherwise.

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