ChatGPT

Transformer-Based Fair Division: Comprehensive Reading List

1. Foundational Theory

- **Fair Division Fundamentals:** Early work on fair division focused on *divisible* resources ("cake-cutting"). A divisible cake can always be split envy-freely among any number of agents, as guaranteed by the Dubins-Spanier moving-knife protocols and later bounded procedures [1]. In contrast, for *indivisible* goods, exact fairness notions like envy-freeness (EF) or proportionality often cannot be achieved [2]. For example, if two agents want a single indivisible item, any allocation leaves one agent empty-handed and envious [3]. This impossibility of perfect fairness without dividing goods motivated *relaxed* fairness criteria and randomized approaches in the indivisible setting [4]. Classic procedures like **Divide-and-Choose** (for two people) and **Selfridge–Conway's** cake-cutting algorithm (for three) illustrate how *continuous division* can guarantee EF, while the jump to discrete items introduces new challenges and the need for approximate fairness. Key fairness notions were formalized in the 1960s-70s – Foley's work on EF and Varian's concept of a competitive equilibrium from equal incomes (CEEI), which yields envy-free and Pareto-optimal allocations when agents can trade divisible shares or use money [5]. These foundations underscore why fair allocation of indivisible goods is both vital (many real resources are indivisible) and non-trivial (classic fair division theorems break down, leading to hardness and impossibility results).

- **Envy-Freeness and its Relaxations:** Envy-freeness (EF) means no one prefers another's allocation [3] – a strong fairness ideal that is often unattainable with indivisible goods. Research therefore turned to relaxed notions. **Envy-freeness up to one good (EF1)**, proposed by Lipton et al. (2004) and others, requires that any envy an agent feels towards another can be eliminated by removing at most one item from the other's bundle [6]. Remarkably, *EF1 can always be achieved* for additive goods: a simple **round-robin** algorithm where agents take turns picking their favorite remaining item guarantees an EF1 allocation in polynomial time [7]. Even though someone might envy another's bundle, the envy is capped to at most one item (hence EF1), making it a very practical notion of fairness. A stronger variant, **EF up to any good (EFX)**, requires that an envy-free state is reached even after removing *any single good* from the envied bundle. EFX is much harder to guarantee; it remains an open question whether an EFX allocation always exists beyond 3 agents. In a breakthrough, Chaudhury et al. (2020) proved EFX *does* exist for 3 agents (with additive valuations) [8], but for 4 or more agents the problem is still unsolved – only partial results under restricted conditions are known (e.g. EFX exists for certain simple valuations or with charity of unallocated items). These relaxations, EF1 and EFX, have become standard fairness benchmarks in lieu of unattainable perfect envy-freeness.

- **Maximin Share (MMS) and Other Fairness Notions:** Another influential concept is the **maximin share (MMS)** fairness criterion, introduced by Budish (2011) as a way to extend the idea of proportionality to indivisible goods. An agent's maximin share is the highest value they can guarantee themselves by partitioning the goods into n bundles (for n agents) and taking the least-valued bundle – essentially a fair share under worst-case division. An allocation is MMS-fair if every agent gets at least their maximin share value. Unfortunately, like EF, an exact MMS allocation need not exist for $n\ge 3$. However, extensive work has gone into *approximate MMS*. For instance, there is always an allocation that gives each agent at least a 2/3 of their MMS value [9], and algorithms have been developed to find such allocations (Amanatidis et al. 2017). Recent breakthroughs have raised this guarantee: Garg & Taki (2021) achieved a $3/4$-MMS approximation, and very recently Akrami & Garg (2023) broke the long-standing 0.75 barrier, showing that one can ensure each agent about 0.75078 of their MMS [10]. These results, while technical, are important because they inch closer to the ideal MMS fairness despite the NP-hardness of exact MMS. Other fairness notions include **Pareto optimality (PO)** – no one can be made better off without hurting someone else – which is about efficiency, and **Nash social welfare (NSW)** – the geometric mean of utilities – which interestingly strikes a balance between fairness and efficiency. In fact, a landmark result showed that *maximizing Nash social welfare yields an EF1 and PO allocation for additive goods* [11]. This ties a fairness notion (EF1) to an economically efficient solution, and though finding the exact NSW-maximizing allocation is NP-hard, it can be approximated or solved for moderate sizes and has guided the design of fair allocation algorithms [12].

- **Impossibility and Complexity Results:** A theme in fair division theory is the balance between what's *desirable* and what's *possible*. On the impossibility side, we know that for indivisible goods, we cannot guarantee even basic fairness and efficiency simultaneously in all cases – for example, no allocation may exist that is both envy-free and Pareto optimal, even with as few as 3 agents and certain preferences (this is related to the famous “EF1+PO vs EFX” gaps and open problems). There are also results like the *Price of Fairness* (PoF) which quantify the loss in total utility when imposing fairness – e.g. ensuring fairness can cap the utilitarian efficiency to a fraction of what an optimal (unfair) allocation would achieve [13]. From a computational perspective, many fair division problems are intractable. Deciding or computing an envy-free allocation (if one exists) is NP-hard in general, as is finding an MMS allocation or optimizing Nash welfare with indivisible goods [14]. Even checking whether a given allocation is EFX can be non-trivial when preferences are not simple additive. These hardness results justify the use of approximation algorithms, heuristics, and more recently machine learning, to find “good enough” allocations in reasonable time. They also motivate relaxed fairness definitions – e.g. EF1 is computationally easier to satisfy than EF, and approximate MMS can be achieved even if exact MMS cannot. In summary, the foundational theory of fair division provides the toolkit of fairness definitions (EF, EF1, EFX, MMS, etc.), classic algorithms for special cases, and an understanding of the limits of what algorithms can do. Any practical system (including a Transformer-based one) must build on these concepts – using the right fairness notion as a target and respecting the known impossibility boundaries.

Key References: Brams & Taylor (1996) on the Adjusted Winner procedure (envy-free, equitable division for two parties) [15]; Foley (1967) for envy-freeness; Steinhaus (1948) for the cake-cutting problem; Varian (1974) on fair division with money (CEEI) [5]; Lipton, Markakis, Mossel, & Procaccia (2004) introducing EF1 [6]; Budish (2011) introducing MMS and approximate CEEI; Caragiannis et al. (2019) linking Nash welfare to EF1 [11]; Procaccia (2016) “Cake Cutting Algorithms” survey for divisible fair division; Bouveret et al. (2016) “Fair Division of Indivisible Goods” survey for an earlier overview.

2. Algorithmic & Computational Social Choice Approaches

- **Constructive Fair Allocation Algorithms:** A rich line of work in computational social choice deals with *explicit algorithms* to allocate indivisible items fairly. One simple yet powerful method is the **Round-Robin (RR) picking sequence**, where agents take turns choosing items in some order. As noted above, RR guarantees EF1 for additive goods [7], and it can be implemented in $O(mn)$ time for m items and n agents. The challenge is deciding the picking order: a fixed arbitrary order works for EF1, but to improve outcomes (like efficiency or envy magnitude) one might choose a clever order. This idea is extended in **alternating random priority** and related methods. Another fundamental procedure is based on the *envy graph*: start by giving out items arbitrarily, then repeatedly find cycles of envy (where A envies B, B envies C, ..., Z envies A) and swap items along the cycle to break the envy loop (this is often called the **envy-cycle elimination** algorithm). Lipton et al. (2004) used this method to prove that an EF1 allocation always exists – the cycle-swapping terminates with no cycles, yielding an allocation where any remaining envy is due to at most one item [6]. These constructive algorithms are important because they not only prove existence of fair allocations (under EF1, etc.) but also provide *polynomial-time methods* to find them. Variants of envy-graph algorithms have been developed for different settings, including chores (where envy-cycle elimination can produce PROP1 – proportionality up to one bad – allocations). Another example is **round-robin with rotations**: agents pick in sequence, but the order of picking may rotate in a way that equalizes opportunities (akin to the draft mechanism in sports). These simple procedures are often used in practice due to their transparency and low computational cost, even if they don't always achieve the absolute best fairness-efficiency tradeoff.

- **Approximation Algorithms for Fairness:** Since many fair division problems are NP-hard, researchers have designed approximation algorithms that guarantee *near-fair* outcomes. We discussed approximation ratios for MMS fairness: Amanatidis et al. (2017) gave the first constant-factor approximation algorithm (guaranteeing each agent $\ge 2/3$ of their MMS) [16]. Their algorithm uses clever item bundling and iterative refinement to ensure no one falls too far below their fair share. This result was significant as it showed we can algorithmically ensure a meaningful fairness guarantee even when exact MMS might fail. Subsequent improvements by Garg, Taki, and others employed more sophisticated "allocation and pruning" techniques to raise the guarantee to 3/4, and very recently slightly above 3/4 [10]. These approximation algorithms often involve solving a series of linear programs or using dynamic programming to assign items in a balanced way. For **EFX**, although we don't know if EFX is always possible, there are algorithms that achieve EFX under additional assumptions – e.g. for *bi-valued* preferences (each item's value is either high or low for each agent) Amanatidis et al. (2021) showed that maximizing Nash social welfare finds an EFX + PO allocation [17], and Garg & Murhekar (2021) even gave a polynomial-time algorithm for that case [17]. These special-case algorithms provide evidence that structured instances of fair division can be solved exactly. Another algorithmic strategy is via **maximum weight matchings or network flows** for certain allocation structures: for example, if each agent can get at most one item (as in one-per-person assignments), an envy-free (even exact EF) allocation can sometimes be found by solving a matching problem under appropriate conditions (this is common in fair bipartite matchings with equal quotas). In more complex scenarios, *mixed integer programming (MIP)* formulations have been used to compute fair allocations for moderate sizes – e.g., one can encode "EF1" as a set of linear constraints and maximize Nash welfare or utility subject to those, then solve with an MILP solver. While worst-case NP-hard, this can find optimal fair solutions for practical instance sizes (up to dozens of agents and items).

- **Parameterized & Heuristic Methods:** Given the complexity of general fair division, researchers also explore parameterized algorithms – algorithms that are efficient if some parameter is small. For instance, if the number of agents n is a small constant, one can brute-force search over allocations (which is exponential in n but that's fine for small n). This is how we can verify things like "does an envy-free allocation exist?" for small groups via exhaustive search. Other parameters could be the number of item *types* (if many items are identical copies, that can simplify the problem) or restricted preference classes (e.g. binary valuations where each item is either "liked" or "not liked" by an agent). Indeed, there are FPT (fixed-parameter tractable) results for certain cases: e.g., for binary additive valuations, an EF1 allocation can be found in polynomial time by reducing to a network flow or scheduling problem. On the heuristic side, local search is a common approach: start with any allocation and then perform exchanges or swaps that improve fairness. This is related to the idea of **envy reduction via swaps** – a heuristic that isn't guaranteed to find an optimal fair allocation but often works well in practice. For example, the algorithm might repeatedly identify the most envious pair of agents and swap some item among them to reduce envy until no further improvement is possible. Such local search heuristics can be guided by fairness metrics or even machine-learning predictions (connecting to Section 3).

- **Fair and Efficient Allocations:** A major focus in recent algorithmic work is achieving *fairness together with efficiency* (typically Pareto optimality or high total utility). Caragiannis et al. (2016/2019) gave a theoretical justification for using the **Maximum Nash Welfare (MNW)** rule: if you compute an allocation that maximizes the product of utilities, it is not only PO by definition (since MNW implies no further mutually beneficial trades can be made), but it also guarantees EF1 for goods [11]. This result, "the unreasonable fairness of Nash welfare," has led to algorithms and heuristics centered on Nash welfare. While solving the MNW exactly is hard, there are approximation algorithms and heuristics (like taking an LP relaxation of the assignment and then rounding it) that get close to MNW. Barman et al. (2018) designed a *pseudopolynomial-time* algorithm to find an allocation that is both EF1 and PO [12] – effectively trying to approximate the MNW solution. It runs in high polynomial time when item values are not too large. No *truly* polynomial-time algorithm is known for EF1+PO (that remains a big open problem), but for a weaker notion like **PROP1** (each agent gets at least 1/n of total value up to one item removed), Barman & Krishnamurthy (2019) did give a polynomial algorithm that also ensures PO [18]. The general message is that fairness and efficiency can sometimes be achieved together, but it often requires careful algorithm design or sacrificing one aspect slightly. For instance, some algorithms use a two-phase approach: first ensure fairness, then adjust allocation (within the fairness constraints) to improve efficiency as much as possible. The **Envy-Free Pareto Improvement** procedure is an example where you start with a fair allocation and see if any trading of items can make someone better off without making others worse off – if yes, you trade (this maintains EF1 or whatever fairness you started with, while increasing utilitarian welfare).

- **Randomized Algorithms and Fair Division with Chance:** When deterministic fair division hits a wall, randomization offers a powerful escape. A seminal concept here is the **Probabilistic Serial (PS)** mechanism for random assignment (Bogomolnaia & Moulin 2001), which allows agents to "eat" a divisible share of each good continuously. PS produces a *fractional allocation* that is envy-free and Pareto optimal in expectation (each agent gets the same fraction of goods as if they had equal entitlement) [5]. While the end result is a lottery over discrete allocations, PS ensures fairness ex-ante. From a computational perspective, one can then sample or round such fractional allocations to get an integral allocation, often preserving fairness approximately (with high probability). Similarly,

Random Serial Dictatorship (RSD) is a simple randomized mechanism: pick a random order of agents and have them pick items in that order. RSD is *ex ante* envy-free and strategy-proof (each agent, not knowing their position, feels they have an equal chance), but any single realization might not be envy-free. Nonetheless, RSD is widely used in practice (e.g. school choice lotteries) because of its simplicity and fairness in expectation. On the algorithmic front, techniques from discrete random processes and approximation algorithms merge in works that, for example, *randomly round LP solutions* to achieve fair allocations with high probability. Randomized rounding can maintain proportionality or EF1 with probability 1 – δ while also giving good total utility. Another angle is **online fair division** (items arriving one by one and must be allocated on arrival): algorithms like competitive equilibria or "envy-fair matching" have been developed using techniques from online algorithms and prophet inequalities, ensuring fairness criteria in a dynamic setting. These are algorithmically interesting because they often require randomness to handle uncertainty in arrival order or future items.

Key References: Lipton et al. (2004) on envy-cycle elimination; Markakis & Psomas (2011) on deterministic EF1 algorithms; Caragiannis et al. (2016 EC / 2019 JACM) on Nash welfare and EF1 [11]; Barman et al. (2018 EC) on EF1+PO algorithms [12]; Barman & Krishnamurthy (2019) on PROP1+PO in polytime [18]; Amanatidis et al. (2017 TALG) on 2/3-MMS approximation; Garg & Taki (2021 FOCS) on 3/4-MMS; Akrami & Garg (2023) on $>3/4$ MMS [10]; Chaudhury et al. (2020 EC) on EFX for 3 agents [8]; Plaut & Roughgarden (2018) introducing EFX; Aziz et al. (2019 IJCAI) on fair allocation of *chores* (negative utilities); Freeman et al. (2020 EC) on equitable allocations. For randomized methods: Bogomolnaia & Moulin (2001) on Probabilistic Serial; Aziz, Caragiannis et al. (2013) on random assignments and their efficiency; Kurokawa et al. (2016) on random allocations as fair lotteries.

3. Machine-Learning Approaches to Fair Division

- **Learning Allocation Mechanisms:** Traditional fair division algorithms are often hand-crafted using human insights. A recent trend is to **learn** allocation policies or mechanisms using data-driven approaches, a paradigm sometimes called *Automated Mechanism Design*. Narasimhan et al. (2016) were early pioneers, proposing a framework to train neural networks to output mechanism decisions (e.g. auction payments) while enforcing properties like incentive-compatibility or fairness [19]. The idea is to parameterize a family of allocation rules by a neural network and then use supervised or reinforcement learning to optimize objectives subject to constraints. In the context of fair division, this means we can attempt to learn a function $f(\text{valuations}) \to \text{allocation}$ that is *guaranteed* to satisfy a fairness notion (by design or via a penalty) and is optimized to meet other criteria (like aligning with past "expert" allocations or maximizing welfare). This approach is especially useful when the "optimal" fair allocation is computationally intractable to compute – instead, the neural network can learn from examples or simulations to allocate in a way that approximates the ideal.

- **Neural Round-Robin and Fairness Learning:** A state-of-the-art example of applying ML to fair division is the **Neural RR (Round-Robin)** model by Maruo et al. (AAMAS 2025) [20] [7]. They focus on achieving EF1 and train a neural network to *choose the picking order* in a round-robin allocation. The model is a Transformer-based architecture that takes reported valuations as input and outputs an ordering of agents (or weights that determine the order). By using a differentiable relaxation of the round-robin process, they can backpropagate through the allocation outcome. The network is trained on example allocations produced by some (possibly implicit) expert rule, with the aim to *mimic* the expert's fairness-conscious decisions while *guaranteeing EF1* by construction (since round-robin yields EF1 regardless of order). Their experiments showed the neural mechanism could outperform static heuristics in terms of certain fairness-efficiency trade-offs, essentially learning a better policy for who picks next. This work demonstrates a template for marrying ML and fairness: design a network that outputs some controllable part of an allocation (like the priority order) and ensure the rest of the allocation process enforces the fairness constraint, then learn the optimal policy within that space. It's a promising direction for making allocation systems more adaptable (learning from data) without sacrificing the guarantees (here, EF1).

- **Neural Combinatorial Optimization (NCO):** Fair allocation can be seen as a combinatorial optimization problem (often NP-hard), and there has been a surge of research on using deep learning to solve or approximate such problems. A landmark in this area was **Pointer Networks** (Vinyals et al. 2015), which introduced a seq2seq model that can output permutations of input elements [21] . Instead of producing a sequence of words, a pointer net produces a sequence of *pointers* to the input (e.g., an ordering of items or tasks). This is directly relevant to allocation – for instance, a pointer network could generate an ordering of item assignments, or a mapping of items to agents by sequentially "pointing" to which agent gets each item in some order. Following pointer nets, **attention-based models** (Transformers) were brought into combinatorial optimization by works like Kool et al. (2019), who showed that a Transformer trained with reinforcement learning can construct near-optimal solutions for routing problems (TSP, VRP) at reasonably large scales [22] . They refer to their model as an *attention model (AM)*, and it significantly outperformed previous sequence models on these tasks, even getting very close to the true optimum on 100-city TSP instances. The success of these models in routing and assignment problems suggests we can apply similar techniques to fair division: e.g., model the allocation as a sequence of item-to-agent assignments and train the network (via RL or supervised learning on solver data) to output high-quality allocations. The RL approach is particularly appealing if we can define a reward that captures fairness (and perhaps utility). For example, an RL reward could be a weighted sum of negative envy and total welfare, encouraging the model to find allocations that are both fair and efficient. The model would then effectively "learn" a heuristic that navigates the huge search space of allocations much faster than brute force. This approach has been hinted at in recent literature, though direct applications to fair division are just emerging.

- **Transformers as Hopfield Networks (Memory for Constraints):** One intriguing development in ML is the realization that Transformer networks can function as **Hopfield networks** (modern continuous Hopfield nets) capable of storing patterns and retrieving them via associative memory [23] . In allocation terms, one can imagine encoding constraints or preferred patterns (like "no agent should get too much more than another") as energy minima in a Hopfield memory, and the Transformer's attention dynamics could naturally enforce or gravitate towards those patterns. For instance, a Hopfield layer could be used to "remember" what a fair allocation looks like (perhaps by storing some known fair allocations as vectors) and guide the model's outputs towards these memories. While this is a more theoretical connection, it underlies some recent architectures (e.g., the Hopfield Pooling in DeepMind's Perceiver model) and could inspire architectures where fairness criteria are hard-coded as attractors in the network's latent space. Practically, one might integrate a Hopfield-like module that, at each step of constructing an allocation, pulls the partial solution towards satisfying envy-freeness or balancedness based on stored prototypes or equations.

- **Learning Auction and Allocation Mechanisms:** Adjacent to fair division is the literature on **learning auctions and mechanisms** with neural networks. For example, *PreferenceNet* by Dütting et al. (NeurIPS 2021) used deep learning to represent and optimize over human preferences in auction design [24]. They trained networks to predict bidders' values and then allocate items and set payments in a way that is approximately incentive-compatible and efficient. Similarly, Curry et al. (IJCAI 2023) and Duan et al. (IJCAI 2023) designed *differentiable auction mechanisms* (for multi-item auctions) using neural networks that ensure certain economic properties (like dominant-strategy incentive compatibility) while maximizing revenue or welfare [24]. Although these works target auctions (which involve payments and typically prioritize revenue), the techniques carry over to pure allocation problems: they use powerful function approximators (neural nets) to learn allocations from data, and enforce constraints by either designing the network's structure or adding penalty terms. In a fair division context, one could similarly train a neural mechanism that takes agents' valuations and outputs an allocation (who gets what) while guaranteeing, say, EF1. The constraint enforcement might use differentiable surrogate metrics for envy or violations (thereby guiding the learning), or enforce structure such as symmetric treatment of agents to promote fairness. An exciting possibility is using *multi-agent reinforcement learning*: imagine each agent is controlled by a neural network policy that bids or negotiates for items, and a central mediator (also a network) learns to adjust the allocation to reach a fair outcome. Such a system could learn to mimic negotiation or price-adjustment processes that humans might use, potentially discovering novel allocation procedures that we haven't analytically described.

- **Partial Decoding and Search Enhancement:** One challenge for neural methods is ensuring the final allocation is *feasible* and fair. Often, a neural network might produce an allocation that slightly violates a constraint (e.g., gives the same item to two agents, or fails a fairness check) if not properly trained or structured. To address this, researchers use techniques like **partial sequence decoding** and search-based refinement. Partial decoding means the model generates an allocation step by step, and at each step, a search (like beam search or Monte Carlo tree search) can explore a few alternatives and validate constraints. For instance, a Transformer could assign items one by one to agents (as a sequence of length m). After assigning, say, 10 out of 100 items, we can pause and evaluate: are any fairness violations emerging (e.g., is one agent getting too much value too early)? If so, a search procedure could adjust the next assignments or even reconsider previous ones (this is like backtracking or repair). Approaches such as **budget constraints integration** (to prevent one agent from hogging too many high-valued items) can be handled by either input features to the Transformer (like remaining "fair budget" for each agent) or by custom decoding rules (e.g., mask out choices that would cause a large envy). This hybrid of neural decoding and constraint-solving is a hot area – it blends the speed and learned intuition of ML with the reliability of constraint programming. In auction literature, something similar is done with *differentiable satisfaction modules* that project neural outputs into the nearest feasible solution (for example, a network proposes an allocation, then a min-cost flow algorithm adjusts it to ensure no capacity constraints are broken). We can imagine a fair division analog: a neural net proposes an allocation which is then "repaired" by an algorithm to satisfy fairness if needed, with the feedback loop training the network to need less repair over time.

In summary, ML approaches to fair division are in their infancy but rapidly growing. They offer the promise of *scalability* (learning heuristics that work for large problems where brute-force fails) and *adaptivity* (improving allocations as they see more instances or changing preferences). However, ensuring fairness constraints are met – a non-negotiable in this domain – means these models must be designed with those constraints in mind (via architecture or training protocol). As research progresses, we expect to see specialized **fair allocation Transformers** that take advantage of symmetry (all agents/items interchangeability), incorporate constraint layers, and leverage training on simulated economies to achieve high-quality fair outcomes. For a graduate student, this is a fertile area to explore, bridging deep learning, combinatorial optimization, and economic fairness.

Key References: Narasimhan et al. (2016) on learning mechanisms with constraints [19]; Shu et al. (2020) "Hybrid Mechanism Design" for combining ML and auctions; Maruo, Takeuchi & Kashima (2025 AAMAS) on Neural EF1 allocation [20] [7]; Dütting et al. (2021 NeurIPS) "PreferenceNet" for neural auction design [24]; Tang (2017) on deep reinforcement learning for resource allocation; Kool, van Hoof & Welling (2019 NeurIPS) on attention models for combinatorial optimization [22]; Vinyals et al. (2015 NIPS) on Pointer Networks [21]; Ramos et al. (2021 ICLR) on Hopfield networks and Transformers [23]; Chen et al. (2022 AAAI) "Pointerformer: Reinforced Multi-Pointer Transformer" for vehicle routing (demonstrating multi-pointer decoding); and OpenAI (2023) on using large language models for optimization (e.g. GPT-4 for scheduling tasks, which hints at using language models for fair division tasks as well).

4. Empirical & Domain Applications

- **House Allocation & Roommate Fairness:** One of the earliest practical fair division problems is the assignment of indivisible goods without money, such as allocating housing or rooms to people. The **House Allocation (Housing Market)** model was solved by Shapley & Scarf (1974) who introduced the **Top Trading Cycles (TTC)** algorithm. In a housing market where each agent initially "owns" one house, TTC finds a *Pareto-efficient and individually fair* outcome by letting agents trade in cycles until no one wants anyone else's house. TTC is also strategy-proof and always produces a *core* allocation (no group can deviate to get a better outcome). While TTC doesn't explicitly aim for envy-freeness, in one-to-one allocation settings it yields outcomes that are "fair" in the sense of respecting everyone's initial endowments and making all voluntary trades. In practice, a common approach for assigning indivisible items (like public housing units or school seats) is **Random Serial Dictatorship (RSD)**, which we discussed earlier: it's essentially TTC when no one owns anything initially (each agent "trades" from an empty endowment, which reduces to random priority selection). RSD has been used in e.g. the allocation of NYC public school seats and certain social housing programs because it's simple and perceived as fair (randomness provides equal opportunity). For the **roommates and rent division** problem – where, say, several roommates want to allocate rooms in a house and split the rent – a classic solution is to find prices for rooms such that each roommate getting their favorite room at those prices yields no envy. This is a form of competitive equilibrium: each roommate pays an equal share of the rent (budget) and buys their most preferred room. It has been shown that an envy-free rent split always exists for reasonable preferences, often found via algorithms that adjust prices (ascending or descending price auctions) until a market-clearing equilibrium is reached. The solution concept ties back to Varian's CEEI [5]. Modern applications include online tools like Spliddit, which implements a rent-splitting algorithm that asks roommates to submit their valuations for each room and then computes an envy-free pricing and allocation. This has seen real-world use, helping people avoid fights over who gets the big bedroom by ensuring that whoever does get it also pays proportionally more rent in a way others agree upon.

- **Course Allocation and Educational Resources:** Allocating seats in courses to students (when demand exceeds supply) is a prime example of fair division in practice. Every year, universities face the problem of assigning students to classes in a way that is fair and efficient, without using prices (typically it's not a cash market). *Fair allocation mechanisms have been deployed*: notably, **Course Match** at Wharton (University of Pennsylvania) is based on the research by Budish (2011) and implemented by Othman, Sandholm, Budish (2010) [25] . In Course Match, students are given artificial budget (points) and price classes through an iterative algorithm that converges to an approximate competitive equilibrium (each student "purchases" a bundle of courses with their budget such that supply = demand). The outcome is approximately envy-free (no student prefers another's schedule given the final implicit prices) and Pareto efficient. This system replaced older first-come-first-served or priority-based methods and was reported to significantly improve student satisfaction. The deployment required solving large linear programs and implementing a truthful elicitation of preferences, illustrating that fairness concepts like *CEEI (with fake money)* can scale to thousands of items and agents in a critical real-world setting. Another domain is *school choice*, where students (or parents) rank schools and a centralized algorithm assigns seats. While the focus there is often on respecting priorities and avoiding envy with respect to priority order (a different formalism), fairness comes into play as *no student should envy another who has lower priority*. Algorithms like TTC and RSD have variants (e.g. Gale-Shapley stable matching or variants of deferred acceptance with fairness constraints) used in school choice programs globally.

- **Spliddit and Online Fair Division Services:** *Spliddit* (spliddit.org) is a free public website that offers implementations of fair division algorithms for various everyday problems [26] . Created by researchers (Goldman & Procaccia 2015), Spliddit has modules for goods division (you input how each person subjectively values each item, and it outputs an allocation plus possibly payments to ensure fairness), rent splitting (as mentioned), and task division. For goods, Spliddit uses the **Adjusted Winner** algorithm for two people (which produces an envy-free, equitable, Pareto optimal split by possibly splitting one item [15]) and a fair division algorithm called the **Maximized Nash Welfare (MNW)** for three or more (which aims to achieve EF1 and high welfare). Studies of Spliddit's usage provide empirical insight: people generally perceive the algorithmic allocations as fair, and the existence of such a tool increases the likelihood that groups find an amicable division. It's a powerful demonstration of theory put into practice – algorithms from papers decades old running in the cloud to solve roommates' and friends' allocation dilemmas. Similarly, the Fair Outcomes website (fairoutcomes.com) offers tools for dispute resolution, using fair division algorithms to settle disagreements (like inheritance division) in a principled way. These platforms not only help people but also gather data that researchers can analyze to understand how humans value fairness.

- **Fair Allocation in Cloud Computing:** In compute clusters and cloud platforms, resources like CPU cores, memory, and GPUs are indivisible units that must be allocated fairly among users or jobs. This gave rise to policies like **Dominant Resource Fairness (DRF)** by Ghodsi et al. (2011) [27] . DRF operates on the principle of *max-min fairness* across multiple resource types: it seeks an allocation where if you look at each user's share of their most demanded resource (their "dominant" resource), those shares are equalized across users [28] [29] . For example, if one job is CPU-heavy and another is memory-heavy, DRF ensures neither job gets more of its bottleneck resource (CPU for the first, memory for the second) than the other in percentage terms. DRF has desirable properties – it's Pareto efficient, strategy-proof (users cannot game by lying about their needs), and it satisfies *sharing incentive* (everyone prefers the allocation to any static equal split) and *envy-freeness* in the sense that no user prefers the resource bundle of another [29] . DRF was implemented in Apache Mesos and Hadoop Yarn, influencing how big data frameworks allocate resources. It exemplifies how fairness notions (here, an analogue of proportional fairness) can be adapted to technical domains. Subsequent research in cluster scheduling examined trade-offs: sometimes strictly fair share can lead to low utilization, so there are schemes that introduce a little slack (for efficiency) while keeping near-fairness, akin to EF1 in spirit (allowing minor inequalities for big efficiency gains). Metrics like the **Jain's fairness index** or **Gini coefficient** are used to measure how balanced the resource allocation is among users. Real-world cluster traces show that fairness policies can greatly affect performance and user satisfaction, making it a crucial aspect of resource management algorithms.

- **Other Domains and Metrics:** Fair division principles have found applications in diverse areas. In **kidney exchange**, while the primary goal is maximizing matches (transplants), there is work on fairness to ensure, for instance, that no demographic group is left behind or that both small and large transplant centers benefit – concepts analogous to fair allocation (though usually handled via constraints or weighted objectives in the optimization). In **spectrum allocation** for wireless networks, auctions are common, but if doing a free allocation (like unlicensed bandwidth), fairness criteria akin to DRF can be applied to give each user a fair share of time or frequency. In **budget allocation** (dividing funds among departments or projects), fairness notions such as proportionality or envy-freeness (no department prefers another's budget) can promote equity, sometimes implemented via participatory budgeting algorithms. These domain-specific problems often introduce unique constraints (geography, legality, etc.), but they still benefit from the core fair division theory.

When evaluating outcomes in such domains, it's common to track *multiple metrics*: e.g., **Utilitarian social welfare** (the sum of utilities) to measure efficiency, **Egalitarian welfare** (the minimum utility) to ensure no one is left too badly off, **Nash welfare** (product of utilities) for a balance, and inequality indices like **Gini** or **Theil index** to quantify disparity. For example, after a course allocation, one might report the percentage of students who got all their top-3 choices (a satisfaction metric), as well as whether any envy situations occurred (fairness), and what the average utility is (efficiency). In cloud computing, aside from fairness index, one measures throughput or job completion time (efficiency). Often there is a trade-off: one can slightly reduce the Gini coefficient of allocation at a modest cost in total throughput – finding that sweet spot is a practical optimization problem.

- **Benchmarks & Datasets:** To facilitate research, some public datasets and benchmarks for fair division have been created. For instance, *PrefLib* is a library of preference profiles that includes some fair-division instances (like utility matrices for agents and items) that can be used to test algorithms. There are also synthetic benchmarks in papers – e.g., randomly generated valuations with certain distributions (uniform, correlated, etc.) to test how algorithms perform under different conditions (like do they remain fair when one agent values everything highly and another has more narrow interests?). In course allocation, anonymized student preference data from certain universities has been used in studies. For fair scheduling, Google's cluster workload traces (which list resource demands of tasks over time) have been used to evaluate fairness algorithms (like how well DRF or its variants perform on real job mixes). The existence of these benchmarks allows for empirical comparison of algorithms and also training data for learned approaches (as in Section 3). Meanwhile, "**social choice tournaments**" and hackathons sometimes include fair division challenges, where different algorithms compete on some fairness and efficiency criteria across a suite of scenarios – this drives home the point that no algorithm is best in all cases, but some robust ones (like round-robin or Nash welfare maximization) tend to perform very well.

Key References: Shapley & Scarf (1974) on one-sided matching (TTC algorithm); Abdulkadiroğlu & Sönmez (1999) on school choice mechanisms; Su (1999) "Rental harmony" (fair rent division solutions); Budish (2011) and Othman et al. (2010 AAMAS) on Course Match [25]; Goldman & Procaccia (2015) on Spliddit [26];

Kurokawa et al. (2015) on fair division user studies; Ghodsi et al. (2011 EuroSys/NSDI) on Dominant Resource Fairness [27] [29]; Parkes et al. (2015) on fair scheduling in cloud; Bertsimas et al. (2011 OR) "Price of Fairness" (formalizing efficiency loss) [13]; Jain's Fairness Index (1984) for networks. Also, Procaccia & Wang (2014) on fair division in practice (an experimental perspective), and recent works on participatory budgeting (e.g., Aziz et al. 2018) which is fair division of money for public projects.

5. Recent Breakthroughs and Open Research (2023–2025)

- **Surveying the Latest Progress:** A must-read recent paper is *Amanatidis et al. (2023)*, **"Fair Division of Indivisible Goods: Recent Progress and Open Questions,"** published in *Artificial Intelligence* [30]. This comprehensive survey (with an extensive reference list) covers advances from roughly the past decade, focusing especially on the last few years' developments in EF1, EFX, MMS, and related topics. It reviews how approximate fairness concepts have been refined (e.g., the introduction of PROP1, EFX), and it outlines the common algorithmic techniques (greedy algorithms, iterative rounding, etc.) used to achieve them [31] [32]. Importantly, each section of the survey highlights *open problems*. For example, it emphasizes that **EFX for general valuations is still open**, that the gap between the best known MMS guarantee (~0.75) and 1 is significant, and that we lack polynomial algorithms for many combined fairness-efficiency objectives. It also points out emerging challenges, like fairness when agents have *unequal entitlements* or *group fairness* (ensuring fairness across groups of agents), and fairness in dynamic or online settings. For a graduate researcher, this survey provides both a state-of-the-art summary and a research agenda – essentially a list of what low-hanging fruit and hard questions remain in fair division.

- **Improved Fairness Guarantees:** In the last two years, we've seen record-breaking results for long-standing problems. One headline result (mentioned above) is by *Akrami & Garg (2023)*, who **broke the 3/4 barrier for MMS** allocations [10]. For years, 0.75 of MMS was the plateau, and some theoretical evidence suggested it might be insurmountable by certain techniques. Akrami & Garg introduced new reduction rules and analysis to get a tiny $\frac{3}{4}+\delta$ fraction of MMS (for an extremely small $\delta\approx0.00078$) for general additive valuations. While this doesn't immediately translate to a practical improvement, it is a significant theoretical breakthrough – it opens the possibility that with more ideas, we could approach closer and closer to full MMS (though 1 *exactly* might never be reachable in polytime unless P=NP). Another result is in the realm of chores (items with *costs*): *Huang & Segal-Halevi (EC 2023)* achieved a **13/11 ≈ 1.1818 approximation for MMS with chores** (meaning each agent's cost is at most 1.1818 times their maximin fair share of bads) [33]. This improved upon the previous 4/3 ≈ 1.333 ratio and is notable because chores are generally harder to allocate fairly than goods. On the flip side, *Li, Wang & Zhou (NeurIPS 2023)* tackled **non-additive preferences for chores** – they found that if agents have submodular cost functions (where costs can have diminishing returns), *no algorithm can guarantee better than a $\min\{n, O(\log m/\log\log m)\}$ MMS approximation* in the worst case [34]. This is a negative result indicating that for more complex preference models, fairness might be fundamentally harder (you can't do constant-factor approximations independent of problem size, unless P=NP). They did show for *subadditive* costs a logarithmic approximation exists [35]. These results show how 2023 has been a year of both pushing *positive* results further (better algorithms) and charting the limits (hardness/inapproximability for generalized settings).

- **EFX and Open Problems on Existence:** The status of **EFX** remains a top open problem, but progress has been made on special cases. In addition to the 3-agent result (Chaudhury et al. 2020) and bi-valued case, a recent paper by *Berger et al. (2021)* made waves by claiming "almost EFX" exists for 4 agents – more precisely, they show an allocation where any envy is bounded by a very small item can be achieved for 4 (and conjecture for more) [36] . This isn't a full EFX (hence "almost full" EFX in the reference), but it suggests we're closing in on understanding the structure for 4 agents. Another development is in combining fairness notions: a 2022 paper by *Farhadi et al.* (AAAI 2022) achieved **EF1 + MMS** simultaneously – they found algorithms that ensure each agent gets EF1 and at least a constant fraction of their MMS. This is notable because most works focused on one fairness criterion at a time. Combining criteria (like EF1 *and* MMS, or EF1 *and* truthfulness, etc.) is a newer direction. We also see more attention to **fair division with *additional constraints***: for instance, fairness in graph allocation (each agent must get a connected subset of items, which is important in dividing land or dividing computational tasks with dependencies) – recent papers in SODA/FOCS have tackled fair path or fair graph partition problems, which are quite complex (EF1 might be achievable but EFX or proportionality become NP-hard to even decide).

- **Interdisciplinary & ML Advances:** On the algorithmic economics side, conferences like *ACM EC (Economics & Computation)* 2023 and 2024 have included papers on fair division with strategic behavior (mechanism design aspects) and papers on empirical evaluation of fair division in labs or field. On the AI/ML side, *NeurIPS 2022/2023* and *AAAI 2023* held workshops and had papers on **learning to allocate**. For example, one NeurIPS 2022 workshop paper used *deep reinforcement learning to fairly allocate healthcare resources over time* (combining ideas of online fair division and RL) [37] . Another NeurIPS 2023 paper applied Graph Neural Networks to an online fair scheduling problem, showing improved fairness-energy tradeoffs over traditional heuristics [38] . These indicate a trend of using modern ML to tackle dynamic or large-scale fair division under uncertainty. *IJCAI 2023* included a paper by *Chan et al.* on fair allocation in donor organ distribution, which is a real-world problem involving both fairness and high stakes efficiency. Moreover, the integration of fairness in broader AI systems (like ensuring large language models or recommendation systems allocate attention fairly across users or tasks) is becoming a topic – e.g., a 2023 IJCAI tutorial "Fairness in Multi-Agent Systems" covered fair division as a component of AI fairness.

- **New Fairness Notions:** The last two years also introduced some new fairness notions and variations. One is **1-out-of-d MMS**, an ordinal approximation: "each agent gets at least one of their top-d bundles in some partition of goods" (proposed by Hosseini et al. 2022) [39] . It tries to capture a more qualitative guarantee than a cardinal fraction. Another notion getting attention is **group fairness**: ensuring that not only individuals, but *coalitions* of agents, feel they got a fair share if they pooled resources. This is much harder (it generalizes envy-freeness to groups) and recent papers have shown impossibility results in general, but have designed algorithms for limited versions (e.g., two groups, or groups of size at most k). The intersection of fair division and *social justice* considerations has led to concepts like weighted fairness (agents have different entitlements or deservingness). For instance, how to fairly allocate aid where each person might have a different claim – new algorithms extend EF1 and MMS to weighted settings, and 2022-2023 saw some progress here (Caragiannis et al. 2022 provided a weighted EF1 allocation existence proof under certain conditions).

- **Tooling and Competitions:** The community has also developed more tools to aid research. For example, there's an open-source library called **公平 (Fair) Division** (hypothetical name for illustration) that implements many known algorithms, making it easier to test and compare them on benchmark instances. And just like SAT-solving or graph algorithm competitions, there has been discussion of a **Fair Division Challenge** where different algorithms (including ML-based) compete on a variety of scenarios for fairness and efficiency scores. If organized, it would likely spur a lot of innovation and practical tuning of algorithms. While not a formal competition yet, some workshops have had "hackathon" style events.

Finally, it's worth noting that fair division is increasingly relevant in public policy (e.g., allocating limited medical supplies, vaccines, etc., fairly among regions – something that came up during COVID-19). Thus, the breakthroughs are not only theoretical: for instance, algorithms for fair vaccine distribution were proposed in 2021-2022, balancing fairness (each country gets its fair share) with urgency (prioritizing where it's needed most). This real-world impact angle means we can expect future breakthroughs to also consider *robustness* (fair allocations under uncertain data), *accountability*, and *stakeholder participation* (mechanisms that involve agents in the allocation process to boost perceived fairness).

Key References: Amanatidis et al. (2023 AIJ) survey [30] ; Aziz et al. (2022 SIGecom Exchanges) survey [40] ; Akrami & Garg (2023) on breaking 3/4 MMS [10] ; Huang & Segal-Halevi (2023 EC) on chore MMS [33] ; Li et al. (NeurIPS 2023) on submodular chore allocation hardness [34] ; Berger et al. (2021) "(Almost) EFX for 4 agents" [36] ; Farhadi et al. (2022 AAAI) on EF1+MMS; Chaudhury et al. (2021 FOCS) on group fairness in division; Lee et al. (2023) on fair division with unequal claims; Kurokawa & Procaccia (2022) chapter on open problems in fair division; plus relevant conference proceedings in STOC/FOCS (e.g., FOCS 2022 had a paper on fair division under complex constraints), NeurIPS/AAAI/IJCAI (2022-2023) for learning-based approaches [37] , and EC/WWW for mechanism design perspectives. These recent works collectively indicate a vibrant and evolving field, with plenty of opportunities for novel research at the intersection of theory, algorithms, and AI.

---

[1] [2] [4] [5] [11] [12] [13] [14] [17] [18] [31] [32] [36] [2202.08713] Algorithmic Fair Allocation of Indivisible Items: A Survey and New Questions
https://ar5iv.org/abs/2202.08713

[3] [6] [7] [9] [16] [19] [20] [24] [25] [26] [40] Learning Fair and Preferable Allocations through Neural Network
https://arxiv.org/html/2410.17500v1

[8] [2002.05119] EFX Exists for Three Agents
https://arxiv.org/abs/2002.05119

[10] [2307.07304] Breaking the $3/4$ Barrier for Approximate Maximin Share
https://arxiv.org/abs/2307.07304

[15] The Adjusted Winner Procedure: Characterizations and Equilibria
https://www.ijcai.org/Proceedings/15/Papers/070.pdf

[21] arxiv.org
https://arxiv.org/pdf/2401.16580

[22] Attention, Learn to Solve Routing Problems! | OpenReview
https://openreview.net/forum?id=ByxBFsRqYm

[23] [Paper] - Hopfield Networks Is All You Need - HTM Forum
https://discourse.numenta.org/t/paper-hopfield-networks-is-all-you-need/7788

[27] [28] [29] Papers
https://mwhittaker.github.io/papers/html/ghodsi2011dominant.html

[30] [2208.08782] Fair Division of Indivisible Goods: Recent Progress and Open Questions
http://arxiv.org/abs/2208.08782

[33] [34] [35] [39] Fair Allocation of Indivisible Chores: Beyond Additive Costs
https://proceedings.neurips.cc/paper_files/paper/2023/hash/aa5d22c77b380e2261332bb641b3c2e3-Abstract-Conference.html

[37] Deep Reinforcement Learning for Efficient and Fair Allocation of ...
https://neurips.cc/virtual/2023/77818

[38] Nonstationary Dual Averaging and Online Fair Allocation
https://openreview.net/forum?id=8bk68fodvD5