Match Me If You Can

The Extension

CSCI 6314
Algorithmic Economics

Sudarshan Damodharan

Department of Applied Mathematics
University of Colorado - Boulder
May 2025

Contents

1 Introduction

Liver transplantation is a life-saving treatment for end-stage liver disease, yet the demand for donor organs far exceeds the supply. In 2021, 13,589 new patients were added to the U.S. liver transplant wait list; that same year, only 9,234 liver transplants were performed [21]. Meanwhile, 1,143 patients died while waiting, and 1,169 were removed from the list after becoming too sick to undergo transplant [21]. Living-donor liver transplantation offers an alternative to waiting for deceased donors: a healthy donor can give a portion of their liver. Unlike kidney exchange, however, a living liver donor must give only part of the organ—either the smaller left lobe or the larger right lobe. The right lobe provides more volume for the recipient but entails significantly higher risk to the donor, whereas the left lobe is safer but may be too small for some patients. This trade-off introduces compatibility and incentive challenges not present in kidney exchange.

Ergin, Sönmez, and Ünver [17] model static two-way liver exchange as a market design problem and propose a precedence-induced adaptive-priority mechanism. They show that their mechanism is individually rational, Pareto-efficient, and incentive-compatible: no participant is worse off by joining, no other feasible arrangement can help someone without hurting another, and participants have no incentive to misreport their medical or preference information. In simulation, this mechanism increases living-donor liver transplants by over 30% while boosting the share of safer left-lobe donations.

While this static model is a breakthrough, real-world liver exchange unfolds dynamically: incompatible or sub-optimal pairs arrive and depart over time, and the decision of when to clear the market can dramatically affect outcomes. Matching every available swap immediately may forgo better future opportunities, but delaying risks patient deterioration, death, or receipt of a deceased-donor organ. Recent work on dynamic matching [19] shows that patient batching can reduce the perishing rate from tens of thousands to dozens in high-volume settings.

1.1 Liver Exchange as a Matching Market

Classical two-sided matching, as introduced by Gale and Shapley, involves two disjoint sets of agents each with strict preferences over members of the other side. Stability is defined by the absence of blocking pairs. Many-to-one variants (college admissions) allow one side to accept multiple matches up to capacity, still coordinated via the deferred acceptance algorithm. One-sided matching (e.g., Shapley–Scarf housing) lets agents trade endowed objects via Top Trading Cycles [11].

Kidney exchange adapts these ideas to organ allocation. Each *pair* consists of a patient and an incompatible living donor. Compatibility -blood and tissue— defines a directed graph in pairs, and feasible *exchanges* are disjoint cycles in which each patient receives a compatible kidney. Under mild assumptions, such as indifference among all compatible kidneys, the Top Trading Cycles and Chains (TTCC) mechanism is Pareto-efficient and strategy-proof, and can be implemented by finding a maximum-cycle cover in polynomial time.

Liver exchange inherits the pair structure of kidney exchange but adds a critical new feature: a living donor can donate either the *safer* left lobe or the *riskier* right lobe [17]. This gives rise to a series of problems:

- *Expanded Compatibility* A donor whose left lobe is too small for a recipient may still donate the right lobe, adding edges to the compatibility graph.
- *Heterogeneous willingness.* Donor–patient pairs differ in their private willingness to accept right-lobe risk.
- *Incentive issues.* Misreporting willingness can manipulate the matching: hiding right-lobe willingness may force a left-lobe match, potentially denying higher-benefit right-lobe swaps to others.

Ergin et al resolve these conflicts by defining a *precedence digraph* on pair types, a two-phase adaptive priority algorithm and deletion operators to ensure each pair commits only to its most preferred feasible exchange when processed.

2 Background: Static Liver Exchanges

Before diving into the liver-exchange mechanism of [17], we first introduce key definitions, notation and axioms of the algorithm.

2.1 Transplant Types and Compatibility and Willingness

Definition 1 (Transplant Type Function)**.** *Let X_i denote the size–and–blood-type vector of patient i, and let Y_j^ℓ, Y_j^r be the corresponding thresholds for donor j's left- and right-lobe grafts. Define*

$$t(j,i) \;=\; \begin{cases} \ell, & X_i \le Y_j^\ell, \\ r, & X_i \not\le Y_j^\ell \quad \text{and} \quad X_i \le Y_j^r, \\ \varnothing, & \text{otherwise}, \end{cases}$$

so that donor j can give the left lobe (ℓ), the right lobe (r), or not at all to patient i.

The function $t(j,i)$ tells us whether donor j can give the left lobe, the right lobe, or no graft at all to patient i, based on their relative sizes and blood-type compatibility. If i fits under j's left-lobe threshold, we label it ℓ; if not but it fits the right-lobe threshold, we label it r; otherwise, no match is possible. This compactly encodes all medical feasibility constraints in one place.

Definition 2 (Compatibility Sets)**.** *The one-way compatibility set is*

$$C(i) \;=\; \{\, j : t(j,i) \neq \varnothing \},$$

and the two-way swap set is

$$D(i) \;=\; \{\, j \in C(i) : \, i \in C(j) \,\}.$$

The set $C(i)$ lists all donors who could give *some* lobe to patient i, ignoring reciprocity. By contrast, $D(i)$ keeps only those j who can also receive from i, so a two-way swap is medically possible. These sets separate "who is physically compatible" from "who can actually trade."

Definition 3 (Willingness and Direct-Transplant Bias)**.** *Each pair i has:*

- *a willingness label $w_i \in \{w, u\}$: w_i=w if i is willing to donate the right lobe, and w_i=u otherwise;*
- *a direct-transplant flag $d_i \in \{m, d\}$: d_i=d if i prefers a direct deceased-donor transplant over any exchange of the same lobe, and d_i=m otherwise.*

Thus the preference relation R_i is one of the four template orders $\{\, R_i^{m/w},\, R_i^{m/u},\, R_i^{d/w},\, R_i^{d/u}\}$

2.2 Mechanism Axioms

Definition 4 (Mechanism Axioms)**.** *A mechanism f maps any reported profile $R = (R_i)_{i\in I}$ to a matching $M \subseteq I \times I$ on the compatibility graph. It must satisfy:*

1. Individual Rationality (IR)*: every i is matched only to a graft it weakly prefers to remaining unmatched (and to direct transplant, if $i \in D(i)$).*
2. Pareto Efficiency (PE)*: no other feasible matching makes some i strictly better off without making any other j worse off.*
3. Incentive Compatibility (IC)*: truthfully reporting (w_i, d_i) is a weakly dominant strategy for each pair.*

2.3 Precedence Digraph

Definition 5 (Precedence Digraph on Pair Types)**.** *Let $T = B \times S$ be the set of patient types and $T_D = B \times S^2$ the set of donor types. Write each pair-type as $X - Y \in T \times T_D$, with*

$$X = (X_1, X_2, X_3) \in T, \quad Y = (Y_1, Y_2, Y_3^\ell, Y_3^r) \in T_D.$$

The precedence digraph on types $(T \times T_D,\ D_\tau)$ *has an edge*

$$(X - Y) \ \to\ (U - V) \quad \iff \quad \begin{cases} X_1 \le V_1,\ X_2 \le V_2,\ X_3 \le V_3^\ell, \\ U_1 \le Y_1,\ U_2 \le Y_2,\ Y_3^\ell < U_3 \le Y_3^r. \end{cases}$$

Intuitively, we can imagine each (X, Y) type as a node and draw an arrow from type $X-Y$ to type $U-V$ exactly when the second donor's left lobe could help the first patient but the first donor's left lobe cannot help the second patient (though its right lobe can). This partial order captures which matches *must* precede others to avoid breaking incentives.

Definition 6 (Precedence Digraph on Pairs)**.** *Given a pool (I, τ) with types $\tau(i) \in T \times T_D$, the* precedence digraph *on I is (I, D) where*

$$i \to j \quad \iff \quad (\tau(i) \to \tau(j)) \in D_\tau.$$

Proposition 1. *The precedence digraph (I, D) is acyclic.*

Sketch. A cycle of length $n \ge 2$ would force a strict increasing chain of patient-size coordinates every two steps, yielding

$$X(0)_3 < X(2)_3 < \cdots < X(n-2)_3 < X(0)_3,$$

a contradiction. Hence no directed cycle can exist and the precdence digraph is acyclic. □

2.4 Deletion and Transformation

Definition 7 (Deletion Operator)**.** *Given a compatibility graph $G' = (I, E')$ and a set $J \subseteq I$ of already-committed pairs, the* deletion operator *for a pair $i \notin J$ does:*

$$A(i \mid J, G') = \big\{ j : \exists\, M \subseteq E',\ M \text{ a matching},\ M(h) \neq \emptyset\ \forall h \in J,\ M(i) = j \big\},$$

$$B(i \mid J, G') = \max_{R_i} A(i \mid J, G'),$$

commit i to some $j \in B(i \mid J, G')$, and update

$$E'' = \big(E' \setminus \{\{i, k\} : k \in I\}\big) \ \cup\ \{\{i, j\} : j \in B(i \mid J, G')\}.$$

Deletion locks in the best achievable partner for i given earlier commitments and removes all inferior edges, guaranteeing that no later step can reassign i to a worse match.

Definition 8 (Transformation Operator)**.** *If pair i is unmatched in $G' = (I, E')$ but is* willing *to donate the right lobe, the* transformation operator *adds all right-lobe edges from i that are individually rational:*

$$E'' = E' \ \cup\ \big\{\{i, j\} \in E_{\mathrm{IR}} : j \neq i,\ t(i, j) = r,\ i \in E_{\mathrm{IR}}(j)\big\}.$$

Transformation defers revealing right-lobe options until a pair cannot match on its safer left lobe, preserving left-lobe priority while enabling additional swaps when necessary.

3 Static Precedence-Induced Adaptive-Priority Mechanism

Having introduced this lengthy set of notation, we can now discuss *static precedence-induced adaptive-priority mechanism.* At a high level the mechanism proceeds in two phases. First, it matches as many left-lobe exchanges as possible in an order that respects the precedence digraph. Then, any unmatched but *willing* pairs have their right-lobe edges revealed and are processed in a secondary priority order.

Definition 9 (Precedence-Induced Adaptive-Priority Mechanism)**.** *Let $G_{\mathrm{IR}} = (I, E_{\mathrm{IR}})$ be the individually rational subgraph, let $\Pi^\ell = (i_1, \ldots, i_K)$ be a topological order of the precedence digraph (I, D), and let Π^r be any secondary order on I. Initialize*

$$G^0 = \big(I,\, E_{\mathrm{IR}} \cap \{\textit{left-lobe edges}\}\big), \quad J^0 = \varnothing, \quad \widetilde{J}^0 = \varnothing.$$

Left-lobe phase: *for $k = 1, \ldots, K$, let $i = i_k$.*

- *If $J^{k-1} \cup \{i\}$ is matchable in G^{k-1}, set $J^k = J^{k-1} \cup \{i\}$, $\widetilde{J}^k = \widetilde{J}^{k-1}$, and apply Deletion to (G^{k-1}, J^{k-1}, i) to obtain G^k.*
- *Otherwise, set $J^k = J^{k-1}$, and if i is willing then $\widetilde{J}^k = \widetilde{J}^{k-1} \cup \{i\}$ and apply Transformation to $(G^{k-1}, \widetilde{J}^k, i)$ to obtain G^k; if i is unwilling then $\widetilde{J}^k = \widetilde{J}^{k-1}$ and $G^k = G^{k-1}$.*

After K steps we have J^K (left-lobe matches) and $\widetilde{J}^K$ (candidates for right lobes).

Right-lobe phase: *enumerate $\widetilde{J}^K = \{i_1^*, \ldots, i_N^*\}$ in order Π^r. Initialize $G_*^0 = G^K$, $J_*^0 = \varnothing$. For $n = 1, \ldots, N$, let $i = i_n^*$:*

- *If $J^K \cup J_*^{n-1} \cup \{i\}$ is matchable in G_*^{n-1}, set $J_*^n = J_*^{n-1} \cup \{i\}$ and apply Deletion on right-lobe edges to obtain G_*^n.*
- *Otherwise set $J_*^n = J_*^{n-1}$ and $G_*^n = G_*^{n-1}$.*

Finally, select any matching M of G_*^N that covers exactly $J^K \cup J_*^N$.

At each left-lobe step, we "lock in" the safest available match for the current pair, removing all its inferior left-lobe options so that later pairs cannot disrupt this choice. If a pair cannot match on the left, we then "transform" it by revealing its right-lobe edges—only those it truly finds acceptable—and defer matching that pair until the second phase. In the right-lobe phase, we again lock in the best available right-lobe partner for each transformed pair.

3.1 Mechanism Properties

Theorem 1 (Correctness of the Static Mechanism)**.** *For any topological order Π^ℓ of the precedence digraph and any secondary priority order Π^r, the precedence-induced adaptive-priority mechanism f_{Π^ℓ, Π^r} produces a matching M that satisfies:*

1. ***Individual Rationality (IR)****: For every pair $i \in I$, either $M(i) = \emptyset$, or $M(i) \in E_{\mathrm{IR}}(i)$.*
2. ***Pareto Efficiency (PE)****: There does not exist another feasible matching M' such that $M'(i) \succeq_i M(i)$ for all i, with $M'(j) \succ_j M(j)$ for some j.*
3. ***Incentive Compatibility (IC)****: Truthful reporting of willingness w_i and direct-transplant bias d_i is a weakly dominant strategy for all $i \in I$.*

Proof of Theorem 1. Below is a sketch of the proofs of the Static Mechanism, individual rationality, Pareto efficiency, and incentive compatibility. Ergin et al, derive a much more through proof of Therorem 1.

Individual Rationality (IR): The mechanism initializes with the graph G^0, containing only left-lobe edges that are individually rational under the pessimistic profile where all pairs are assumed unwilling. At each iteration of the left-lobe phase, the Deletion operator selects for a pair i one of its best achievable partners (if one exists) and prunes all worse edges. Importantly, all such matches lie within E_{IR}, the individually rational edge set. In the right-lobe phase, edges added via the Transformation operator are also restricted to those in E_{IR}. Hence, any pair that becomes matched through either phase receives a graft that it weakly prefers to remaining unmatched (and, if applicable, to direct transplant), and all unmatched pairs are those for whom no such graft was available. Therefore, the output matching M is individually rational.

Pareto Efficiency (PE): For each pair i that becomes committed (either in the left-lobe or right-lobe phase), the mechanism ensures that i is matched to one of its best achievable partners given the current graph and all previously committed pairs. The Deletion operator then removes all inferior edges for i, ensuring no strictly better match becomes feasible later. Because the graph only shrinks over time—either through deletion or irrevocable commitments—it is impossible for a Pareto improvement to arise since no unmatched pair is left out while a preferred match remains feasible, and no matched pair can be rematched to a better partner without harming someone else. Hence, M admits no Pareto improvement and is therefore Pareto efficient.

Incentive Compatibility (IC): Each pair's strategic control is limited to their reported willingness w_i and direct-transplant bias d_i. Falsely reporting unwillingness suppresses access to the Transformation operator and the right-lobe edges, possibly leaving the pair unmatched despite better feasible matches. Since the Deletion operator always matches a pair to one of its best achievable options given current commitments, misreporting willingness cannot result in a strictly better partner. Likewise, misreporting direct-transplant bias cannot cause a strictly better direct match to be chosen, since such matches are included in the initial graph and evaluated alongside all others. Thus, truth-telling is a weakly dominant strategy, and the mechanism is incentive compatible. □

4 Dynamic Matching

Having established the static precedence-induced mechanism and its properties, we now turn to the question of *when* to clear the market. In practice, liver-exchange pools evolve over time as new patient–donor pairs arrive, existing pairs depart, and compatibility relations change. Static, one-shot clearing algorithms cannot capture the trade-off between waiting longer to thicken the pool (and hence increasing the chance of finding high-cardinality exchanges) versus matching sooner to reduce mortality on the waitlist. *Dynamic matching* models this temporal dimension explicitly and leads to policies that balance throughput, fairness, and waiting-time objectives.

In 2016 Akbarpour et al, in their paper "Thickness and Information in Dynamic Matching Markets" [19] derive a mechanism for a dynamic matching market where agents arrive and depart stochastically; they show that waiting to thicken the market when imminent departures are observable substantially reduces the fraction of unmatched agents, whereas greedy matching is nearly optimal when departure times cannot be identified.

4.1 Dynamic Policies

We model the liver-exchange pool as a continuous-time stochastic system in which patient–donor pairs arrive, depart, and are matched.

Let

$$\{T_k\}_{k\geq 1} \quad \text{with} \quad T_k = \inf\{t : N(t) = k\}$$

be the arrival times of a Poisson process of rate $\lambda > 0$. Upon arrival at time T_k, pair k enters the pool. Each pair i that remains unmatched incurs an independent $\mathrm{Exp}(\mu)$ departure clock. Therefore, the pool size

$$N(t) = \#\{\, i : T_i \leq t \text{ and } i \text{ has not departed or been matched by } t\}$$

evolves as a birth–death process with birth rate λ and death-rate $\mu\, N(t)$. In a steady state $N(t) \sim \mathrm{Pois}(\lambda/\mu)$.

At each time t, conditional on $N(t) = n$, we denote the set of active pairs by $V(t)$. We assume that blood-type and size compatibilities induce an Erdős–Rényi graph

$$G(t) = (V(t), E(t)), \qquad \Pr\big(\{i,j\} \in E(t) \mid V(t)\big) = p, \quad \text{independent across } i < j,$$

where $p \in (0,1)$ captures the marginal probability that two randomly selected pairs can exchange a lobe (left or right).

The planner observes the history (filtration) $\mathcal{F}_t$ generated by arrivals, departures, and revealed compatibilities up to time t and must choose a sequence of clearing times

$$0 = \tau_0 < \tau_1 < \tau_2 < \cdots$$

adapted to $\{\mathcal{F}_t\}$. At each τ_k:

- The current pool is $V(\tau_k)$ with compatibility graph $G(\tau_k)$.
- The planner runs a static matching mechanism f (the precedence-induced algorithm) on $G(\tau_k)$.
- All matched pairs are removed from the pool; the remainder continue under the same arrival–departure dynamics.

Having introduced the notation used in the dynamic matching problem, we can discuss the policies used by [19].

4.2 Greedy Matching Policy

We begin by introducing the *greedy* policy formally and state its main performance guarantee.

Definition 10 (Greedy Policy)**.** *Let arrivals occur at times $T_1 < T_2 < \cdots$ of a Poisson(λ) process, and let $V(t)$ and $G(t) = (V(t), E(t))$ be as in Section 4.1. The* greedy *policy clears the market at every arrival time:*

$$\tau_k = T_k,$$

and executes the following step immediately upon arrival of agent $i = T_k$:

1. *Reveal all edges $\{\{i,j\} : j \in V(\tau_k^-)\} \subseteq E(\tau_k)$.*
2. *If there exists any unmatched neighbor $j \in V(\tau_k^-)$ with $\{i,j\} \in E(\tau_k)$, choose one arbitrarily and set*
$$M(\tau_k) = M(\tau_k^-) \cup \big\{\{i,j\}\big\},$$
then remove i and j from the pool.
3. *Otherwise, leave i unmatched:*
$$M(\tau_k) = M(\tau_k^-), \quad V(\tau_k) = V(\tau_k^-) \cup \{i\}.$$

All other times $t \neq \tau_k$ no matching occurs, and agents depart according to their $\mathrm{Exp}(\mu)$ *clocks if still unmatched.*

Proposition 2. *Under the greedy policy, the clearing times are exactly the arrival epochs:*

$$\tau_k = T_k, \quad k = 1, 2, \ldots.$$

Meaning no agent ever waits for a scheduled clearing beyond the next arrival—matches occur immediately upon each arrival event.

Proposition 3. *Let D_i be the indicator that arriving agent i eventually departs unmatched under greedy. In steady state,*

$$\Pr[D_i = 1] \;=\; \frac{\mu}{\lambda p \;+\; \mu}.$$

Consequently, in very "thick" market $\lambda p \gg \mu$, the perishing probability decays as $O\big(\frac{\mu}{\lambda p}\big)$, whereas in sparse markets $\lambda p = O(\mu)$ it remains bounded away from zero $\Theta(1)$), meaning a non-trival fraction still fail to match.

Theorem 2 (Constant-Factor Approximation to the Uninformed Optimum)**.** *Let Π^* be any policy that does not observe individual departure clocks, and let*

$$M_G(T),\ M^*(T)$$

be the total matches up to time T under the greedy policy and Π^*, respectively. There exists a constant $C = C(p/\mu) < \infty$ such that for all $\lambda, \mu, p > 0$ and all $T > 0$,

$$\mathbb{E}\big[M^*(T)\big] \ \le\ C\,\mathbb{E}\big[M_G(T)\big].$$

Hence greedy achieves a constant-factor approximation to the best policy without departure-time information [19].

4.3 Patient (Threshold) Policy

Although the greedy policy matches agents immediately and attains a constant-factor approximation to the uninformed optimum (Theorem 2), its steady-state perishing probability

$$\Pr[D_i = 1] \ =\ \frac{\mu}{\lambda p + \mu}$$

remains bounded away from zero whenever $\lambda p = O(\mu)$. To drive the fraction of unmatched agents down toward zero, we can instead allow the pool to accumulate until its size reaches a predetermined threshold m, then clear all matches at once. This *threshold-m* approach trades a batching delay of order m/λ for an exponentially small perishing probability $O(e^{-pm/4})$.

Definition 11 (Threshold-m Policy)**.** *Fix a threshold $m \in \mathbb{N}$. Let*

$$\tau_0 = 0, \qquad \tau_k = \inf\{\, t > \tau_{k-1} : N(t) = m\},$$

where $N(t) = |V(t)|$ is the pool size. At each clearing time τ_k, the planner:

1. *Observes $G(\tau_k) = (V(\tau_k), E(\tau_k))$.*
2. *Runs a matching subroutine on all of $V(\tau_k)$.*
3. *Removes all matched pairs, leaving $V(\tau_k^+)$ as the unmatched survivors.*

No matching occurs at times $t \neq \tau_k$, and unmatched agents depart according to their $\mathrm{Exp}(\mu)$ *clocks.*

Proposition 4 (Expected Clearing Interval)**.** *Under the threshold-m policy, the expected time until the first clearing satisfies*

$$\mathbb{E}[\tau_1] \ \le\ \frac{m}{\lambda} \ +\ \frac{1}{\mu}\,,$$

and hence the typical waiting time for an agent is $O(m/\lambda)$.

Proposition 5 (Exponential Decay of Perishing)**.** *Let D_i indicate that agent i departs unmatched. In steady state,*

$$\Pr[D_i = 1] \ \le\ \exp\!\Big(-\tfrac{p\,m}{4}\Big).$$

Consequently, by choosing $m = \frac{4}{p}\ln(1/\delta)$, the perishing probability can be driven below δ, at the cost of increasing the expected wait to $O\big((\ln(1/\delta))/\lambda\big)$.

Sketch. Once the pool reaches size m, its composition is an i.i.d. sample of m arrivals. A standard coupon-collector–type argument shows that the probability no compatible partner exists for a given agent decays like $\exp(-pm/4)$. The bound on $\mathbb{E}[\tau_1]$ follows by comparing the arrival-only hitting time m/λ plus an additional $O(1/\mu)$ for the last agent's departure clock. □

In the threshold-m policy, agents typically wait on the order of m/λ time units before a matching is attempted—unlike the greedy policy, which matches instantly. This deliberate batching delay comes with a dramatic benefit: the perishing probability falls from a constant $\Theta(1)$ under greedy to an exponentially small $O(e^{-pm/4})$ when using threshold m. Consequently, by tuning m, their is a tradeoff between a linear increase in average wait time for an exponential decrease in the fraction of unmatched agents.

4.4 Patient(α) Policy

The threshold-m policy trades batching delay against perish reduction by waiting for a fixed pool size, but it does not adapt to the actual waiting times of individual agents. To blend both head-count and time considerations, we introduce the patient(α) family of policies, which clears the market whenever the pool is "thick enough" or any agent has waited too long.

Definition 12 (Patient(α) Policy)**.** *Fix a head-count threshold $m \in \mathbb{N}$ and a time-scale parameter $\alpha > 0$. Let T_i be the arrival time of pair i, and define*

$$a_i(t) \;=\; t - T_i, \qquad N(t) \;=\; \big|V(t)\big|,$$

where $V(t)$ is the set of unmatched pairs at time t. We recursively define clearing times $\{\tau_k\}_{k\geq 0}$ by

$$\tau_0 = 0, \quad \tau_k = \inf\Big\{\, t > \tau_{k-1} : N(t) \geq m \;\vee\; \max_{i \in V(t)} a_i(t) \geq \tfrac{\alpha}{\lambda} \Big\}.$$

Between these epochs, pairs arrive as a Poisson process of rate λ and (if unmatched) depart with independent $\mathrm{Exp}(\mu)$ *clocks. At each clearing time τ_k:*

(i) Observe the current pool $V(\tau_k)$ and compatibility graph $G(\tau_k) = (V(\tau_k), E(\tau_k))$.

(ii) Compute the matching $M_k = f\big(V(\tau_k), E(\tau_k)\big)$ using the static mechanism f.

(iii) Remove all pairs in M_k from the pool; the survivors $V(\tau_k^+) = V(\tau_k) \setminus M_k$ continue with their existing departure clocks.

Immediately after τ_k, resume monitoring $N(t)$ and $\{a_i(t)\}_{i \in V(t)}$ for $t > \tau_k$ and repeat the above to define τ_{k+1}.

The scalar α tunes the relative importance of waiting time versus market thickness.A large α delays clearing until agents endure longer sojourns, pushing the policy toward a pure threshold-m behavior. While a smaller α forces earlier clears when any agent's wait exceeds α/λ, resembling a time-driven version of the greedy policy. We can can calibrate α based on the rates λ, μ and compatibility p to navigate the perish-vs-wait trade-off.

For a suitably chosen α, the patient(α) policy achieves a perishing probability and average wait time that jointly lie within constant factors of the optimal trade-off curve—without needing full knowledge of the global graph structure. In particular, it interpolates smoothly between the exponential perish-rate decay of the threshold policy and the zero-delay guarantee of the greedy policy.

4.5 From Dynamic Matching to Dynamic Liver Exchange

To introduce the dynamic timing rules into the liver exchange setting, we simply replace the generic matching subroutine f by the static precedence-induced mechanism f_{Π^ℓ,Π^r}. Concretely, at each clearing epoch τ_k defined by any of the Greedy, Threshold, or Patient(α) policies, we execute

$$M_k = f_{\Pi^\ell,\Pi^r}\big(V(\tau_k), E(\tau_k)\big)$$

on the current pool and remove the matched pairs, just as in the generic model.

When grafting the dynamic timing rules onto the static precedence-induced mechanism, we have to ensure that at each clearing epoch τ_k the compatibility graph $E(\tau_k)$ is built using the Deletion-Transformation procedure. Consequently, a head-count threshold m must measure the "effective" pool size by counting only those pairs whose left-lobe opportunities have been exhausted, and the Patient(α) policy must track each agent's sojourn time alongside its transformation status, so that only fully processed pairs contribute to the triggering condition.

Moreover, private willingness and donor-risk heterogeneity demand adaptive thresholds. In an age-based policy, one can individualize the time threshold by setting $\alpha_i \propto 1/\mathrm{risk}_i$, causing more risk-averse donors to trigger an earlier clearing. Similarly, a fixed head-count rule can be extended to a two-dimensional threshold (m^ℓ, m) that requires at least m^ℓ left-lobe-only pairs before any right-lobe matches are executed.

5 Extension: Reinforcement Learning & Markov Descion Proccesss for Dynamic Clearing

These timing policies adapt only a small number of parameters, a full *reinforcement learning* (RL) framework can discoverstate-dependent clearing policies that optimize a long-run objective combining match volume, waiting costs, and donor-risk. By casting the timing decision as a Markov decision process (MDP), an RL agent can learn to exploit subtle patterns in the evolving pool—such as varying age distributions, compatibility heterogeneity, and seasonality in arrivals—that fixed head-count or age-based heuristics can't capture.

5.1 MDP Formulation

We can cast the dynamic clearing decision as a discrete-time Markov Decision Process (MDP) whose state updates occur at fixed intervals $\Delta > 0$. Between decision times, the pool evolves under the continuous-time arrival–departure process [22].

Definition 13 (State Space)**.** *Let*

$$\mathcal{S} = \Big\{ s = \big(n_\ell, n_r, n_w, \bar{a}, \bar{r}\big) \Big\}$$

where

- *n_ℓ and n_r are the numbers of unmatched pairs currently eligible for left- and right-lobe matching, respectively,*
- *n_w is the count of "transformed" (willing) pairs,*
- *$\bar{a}$ is the maximum sojourn time $\max_i a_i$,*
- *$\bar{r}$ is the average donor-risk level in the pool.*

Definition 14 (Action Space)**.**

$$\mathcal{A} = \{\mathsf{Wait}, \mathsf{Clear}\}.$$

At each discrete decision epoch k, the agent chooses $A_k \in \mathcal{A}$.

Definition 15 (State-Transition Kernel)**.** *Let $\phi : \mathcal{S} \to \mathcal{S}$ be the deterministic "clear-and-remove" map:*

Let $P_{\text{evo}}(s' \mid s, \Delta)$ denote the probability that, under no clearing, the continuous-time birth–death process of arrivals (rate λ) and departures (rate μ) transitions from s to s' over a time interval of length Δ. Then the MDP transition probabilities are

$$P(s' \mid s, a) = \begin{cases} P_{\text{evo}}(s' \mid s, \Delta), & a = \mathsf{Wait}, \\ P_{\text{evo}}\big(s' \mid \phi(s), \Delta\big), & a = \mathsf{Clear}. \end{cases}$$

In particular, if $a = \mathsf{Clear}$, collapse s to $\phi(s)$ by running the precedence-induced mechanism, then let the pool evolve for Δ.

Definition 16 (Reward Function)**.** *For a transition $s \xrightarrow{a} s'$, define the immediate reward*

$$r(s, a, s') = \big|\, f_{\Pi^\ell, \Pi^r}(s)\big| - \beta \,\big|\textit{perished in } \Delta\big| - \gamma \,\big(\textit{risk-cost of right-lobe matches}\big),$$

where $\beta, \gamma > 0$ trade off throughput, perishment, and donor-risk. Here $\big|f_{\Pi^\ell, \Pi^r}(s)\big|$ is the number of matches executed if $a = \mathsf{Clear}$, and zero if $a = \mathsf{Wait}$.

Definition 17 (Discounted Objective)**.** *With discount factor $\rho \in (0, 1)$, the goal is to find a policy $\pi : \mathcal{S} \to \mathcal{A}$ maximizing*

$$J^\pi(s_0) = \mathbb{E}^\pi \Big[\sum_{k=0}^{\infty} \rho^k \, r(S_k, A_k, S_{k+1}) \,\Big|\, S_0 = s_0 \Big].$$

To solve the MDP defined above, we can employ two broad families of reinforcement-learning methods: Q-learning and policy optimization.

Q-learning, seeks to approximate the optimal action-value function $Q^*(s, a)$. At each decision epoch k the agent observes state S_k, selects action A_k (either Wait or Clear) according to an exploration policy, and receives reward $r_k = r(S_k, A_k, S_{k+1})$. This update rule drives Q toward the true action-value function [22].

A second approach is policy optimization via gradients. We can parametrize a stochastic policy $\pi_\theta(a \mid s)$ and define the objective

$$J(\theta) \;=\; \mathbb{E}\Big[\sum_{k=0}^{\infty} \rho^k \, r(S_k, A_k, S_{k+1})\Big],$$

where $A_k \sim \pi_\theta(\cdot \mid S_k)$.

Using a gradient estimator or actor–critic methods, we can compute unbiased or low-variance estimates of $\nabla_\theta J(\theta)$ and update θ accordingly [22].

In either approach, we enforce that whenever the policy selects Clear, the matching executed is exactly the static precedence-induced mechanism f_{Π^ℓ, Π^r}. By incorporating this constraint—either by rejecting or heavily penalizing any policy output that attempts to clear outside this mechanism—we preserve individual rationality, Pareto efficiency, and incentive compatibility at every batch.

5.2 Advantages

Unlike fixed threshold or age-based heuristics, an RL agent learns a fully state-dependent policy $\pi^*(s)$ that can exploit the entire vector of pool statistics $(n_\ell, n_r, n_w, \bar{a}, \bar{r})$. For example, it may choose to clear even before a head-count threshold is reached if the average donor risk $\bar{r}$ spikes, or it may defer clearing in a thick pool with favorable composition to accumulate further advantages. Moreover, by continuously updating π_θ based on observed transitions, the RL policy adapts automatically to changes in arrival rate λ, departure rate μ, compatibility probability p, and urgency distributions, eliminating the need for manual retuning of parameters m or α. Finally, since the RL objective $J(\theta)$ directly encodes the long-run trade-off between match volume, waiting costs, and donor-risk penalties, the learned policy achieves a holistic optimization that balances these competing goals more effectively than separate, sequential heuristics.

6 Conclusion

In this extension, we first reviewed the static precedence-induced adaptive-priority mechanism of Ergin,et al, which guarantees individual rationality, Pareto efficiency, and incentive compatibility in a liver-exchange pool where each donor may give either a left or right lobe. We then introduced a suite of dynamic timing rules—including greedy matching, threshold-based batching, and the Patient(α) family—that leverage stochastic arrival–departure models to trade off waiting times, match volume, and donor risk. By grafting the static mechanism into these dynamic policies, we can obtain a fully incentive-compatible procedure that can be tuned to balance safety with efficiency.

Looking forward, a research direction is to cast the clearing decision as a Markov decision process and apply reinforcement-learning methods to discover non-threshold, state-dependent policies. By defining each state to capture the pool's composition, sojourn-time distribution, and aggregate donor-risk, and by specifying rewards that jointly penalize perishing, waiting, and risk, an RL agent can learn a dynamic clearing strategy that maximizes long-run welfare. Such an approach can adapt automatically to shifts in arrival rates, compatibility patterns, and patient-level heterogeneity—while preserving the IR/PE/IC guarantees of the static mechanism by always invoking it at each batch—thereby offering a robust, data-driven clearing rule that outperforms fixed heuristics in complex, evolving liver-exchange environments.

References

[1] Gale, D., and Shapley, L. S. *College Admissions and the Stability of Marriage.* The American Mathematical Monthly, 69(1), 9–15, 1962.

[2] Pass, R., and Rabin, M. *A Course in Networks and Markets.* , Cornell University, 2020.

[3] Alvin E Roth, "The college admissions problem is not equivalent to the marriage problem," *Journal of Economic Theory*, vol. 36, no. 2, pp. 277–288, 1985. doi: 10.1016/0022-0531(85)90106-1.

[4] Atila Abdulkadiroğlu and Tayfun Sönmez, "Matching Markets: Theory and Practice," Manuscript, Duke University & Boston College, n.d.

[5] AlvinE.Roth, "Deferred Acceptance Algorithms: History, Theory, Practice, and Open Questions," *International Journal of Game Theory*, vol. 36, no. 3, pp.537–569, 2008. doi: 10.1007/s00182-008-0147-1.

[6] Lentine KL, Smith JM, Lyden GR, Miller JM, Booker SE, Dolan TG, Temple KR, Weiss S, Handarova D, Israni AK, Snyder JJ. OPTN/SRTR 2023 Annual Data Report: Kidney. *American Journal of Transplantation.* 2025;25(2S1):S22–S137. doi:10.1016/j.ajt.2025.01.020.

[7] Organ Procurement and Transplantation Network (OPTN) and Scientific Registry of Transplant Recipients (SRTR). *OPTN/SRTR 2023 Annual Data Report.* U.S. Department of Health and Human Services, Health Resources and Services Administration; 2025. Available from: `https://srtr.transplant.hrsa.gov/annualdatareports`.

[8] United Network for Organ Sharing (UNOS). UNOS National Data: Transplant Activity and Waiting List. Accessed July 2024. Available from: `https://optn.transplant.hrsa.gov/data/view-data-reports/national-data/`.

[9] Scientific Registry of Transplant Recipients (SRTR). Chapter 1: Kidney Transplantation. In: *OPTN/SRTR 2023 Annual Data Report.* U.S. Department of Health and Human Services; 2025. Available from: `https://srtr.transplant.hrsa.gov/annualdatareports/chapters/kidney_transplantation.aspx`.

[10] Lentine KL, Smith JM, et al. Trends in Kidney Transplantation, 2013–2023. *American Journal of Transplantation.* 2025;25(2S1):S22–S137. doi:10.1016/j.ajt.2025.01.020.

[11] Lloyd S. Shapley and Herbert Scarf, "On Cores and Indivisibility," *Journal of Mathematical Economics*, vol. 1, no. 1, pp. 23–37, 1974. doi:10.1016/0304-4068(74)90033-0.

[12] Atila Abdulkadiroğlu and Tayfun Sönmez, "House Allocation with Existing Tenants," *Journal of Economic Theory*, 1999. doi:10.1006/jeth.1999.2566.

[13] Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver, "Kidney Exchange," *Quarterly Journal of Economics*, May 2004. doi:10.1162/0033553041382157.

[14] Alvin E. Roth, Tayfun Sönmez, and M.Utku Ünver, "Pairwise Kidney Exchange," *Journal of Economic Theory*, Dec.2005. doi:10.1016/j.jet.2005.02.002.

[15] Alvin E. Roth, Tayfun Sönmez, and M.Utku Ünver, "Efficient Kidney Exchange: Coincidence of Wants in Markets with Compatibility-Based Preferences," *American Economic Review*, Jun. 2007

[16] M.Utku Ünver, "A Kidney Exchange Clearinghouse in New England," *American Economic Review Papers and Proceedings*, May 2005

[17] Haluk Ergin, Tayfun Sönmez, and M. Utku Ünver "EFFICIENT AND INCENTIVE-COMPATIBLE LIVER EXCHANGE", *Econometrica*

[18] Ashlagi, I. and M. Ünver (2010). "Dynamic Kidney Exchange," *Operations Research*, **58**(5), 1439–1453.

[19] Akbarpour, M., S. Li and S. O. Gharan (2020). "Thickness and Waiting in Dynamic Matching Markets," *Proceedings of the ACM on Economics and Computation*, **8**(2), 1–35.

[20] Ünver, M. U. (2010). "Dynamic Kidney Exchange," *Management Science*, **56**(3), 606–620.

[21] Organ Procurement and Transplantation Network (OPTN) and Scientific Registry of Transplant Recipients (SRTR). OPTN/SRTR 2021 Annual Data Report. U.S. Department of Health and Human Services, Health Resources and Services Administration; 2023.

[22] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.