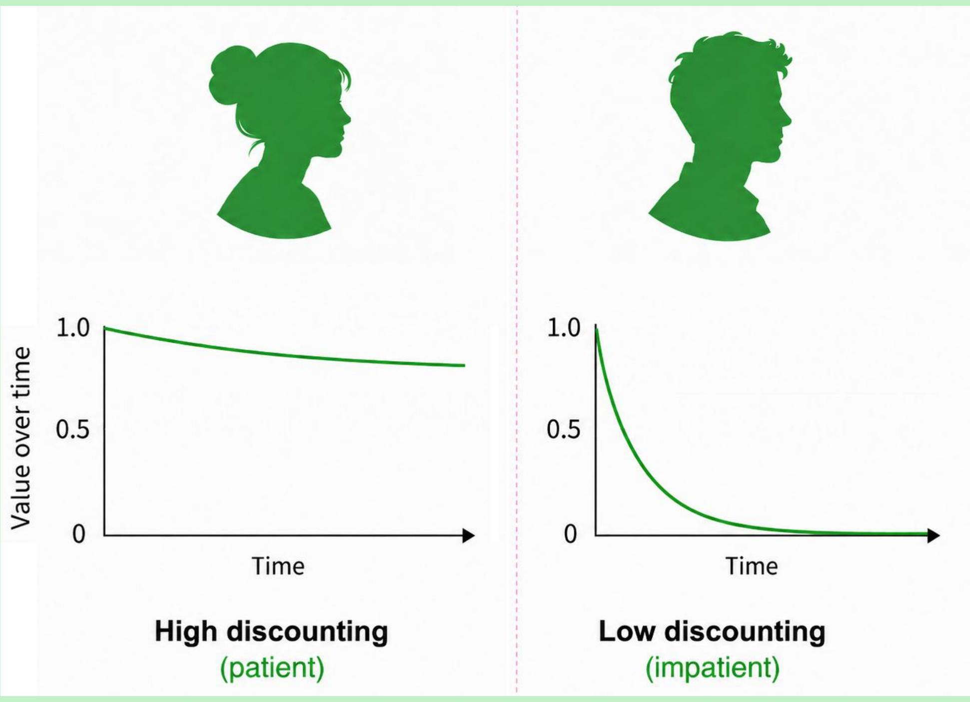
Social Welfare under Heterogeneous Time Preferences

Sarvin Bahmani, Soumyajit Paul, Sven Schewe, Shadi Kalat, Ashutosh Trivedi.

Heterogeneous Time Preferences



Principal$_i$ payoff under policy π:

$$V_i^\pi = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \lambda_i^t R_i^t \right]$$

Social Welfare: $\mathrm{SW}(\pi) = \sum_i V_i^\pi$

Optimisation Objective: $\max_\pi \mathrm{SW}(\pi)$

Long-term Behavior: Lexicographic Tail

M_0 original MDP → M_1 optimal for P_0 → M_2 tie-break P_1 → ... → M_n long-term tail

Order principals by patience:

$$\lambda_0 > \lambda_1 > \lambda_2 > \lambda_3 > \cdots > \lambda_n$$

1. Optimise MDP for most patient principal. Restrict to optimal actions.
2. Break ties using the next most patient principal.
3. Continue until last principal.

After a finite horizon, deviations cannot help

for every deviation, there exists a finite horizon *k* such that, for all $t \geq k$: $\Sigma_i \, \Delta_t(s, a, i) \geq 0$

Why κ is small under spacing?

- A typical bound has shape $\mathrm{k} = \mathcal{O}\left(\frac{\log(\text{value ratio})}{\log(\lambda_i/\lambda_{i+1})}\right)$
- Only the prefix before *k* needs time-dependent optimisation; after that, follow the long-term tail.

Deviation rewards: $\boldsymbol{R_k^\Delta(s, a) = \sum_i \lambda_i^k \, \Delta_0(s, a, i)}$

Initial Deviation

for Principal$_i$:

$$\Delta_0(s, a, i) = R(s, a, i) + \lambda_i \sum_{s'} p(s' \mid s, a) V_i(s') - V_i(s)$$

after t steps

$$\Delta_t(s, a, i) = \lambda_i^t \Delta_0(s, a, i)$$

social effect

$$\sum_i \Delta_t(s, a, i) = \sum_i \lambda_i^t \, \Delta_0(s, a, i)$$

original MDP → long-term tail → flipping point → unroll to DAG → backward induction

$$\pi^* = \underbrace{[\text{finite prefix}]}_{\text{counting memory}} \cdot \underbrace{[\pi_{\text{long}}^{\omega}]}_{\text{stationary tail}}$$

Complexity Contrast

Insist on stationary simplicity	Allow structured finite memory
▪ Loses social welfare	✓ Attains optimal social welfare.
▪ Positional threshold is NP-hard	✓ Welfare computable in PTIME.
▪ Randomisation help but not optimum	✓ Pure counting strategies are sufficient.

Scalability