League of Legends Ban–Pick as a Min-Cost Flow Problem

Hanbyul (Han) Lee (he/him)

PhD Student, Applied Mathematics
Department of Mathematical and Statistical Sciences
University of Colorado Denver

November 30, 2025

hanbyul.lee@ucdenver.edu

Draft pick, but make it linear programming.

Motivation & Outline

Motivation

- League of Legends ban–pick is a constrained combinatorial decision problem.
- We have:
  - Roles (TOP, JGL, MID, ADC, SUP),
  - A pool of champions,
  - Utilities for each role–champion combination.
- Question: *Given utilities and constraints, what is the optimal lineup?*

Outline

1. Min-cost flow (primal) recap
2. LoL ban–pick as a min-cost flow model
3. Parametric analysis of champion utilities
4. Python implementation (primal & parametric)
5. Brief idea: dynamic/game-theoretic extension

What a LoL Ban–Pick Screen Looks Like

- Roles on the left (TOP, JGL, MID, ADC, SUP), enemy team on the right.
- Each team builds a lineup by choosing one champion per role, with no duplicates.
- Our model abstracts this screen into a min-cost flow problem.

Setting and Parameters

Discrete structure

- Role set $R = \{\text{TOP}, \text{JGL}, \text{MID}, \text{ADC}, \text{SUP}\}$, $|R| = 5$.
- Champion set $C \subseteq C_{\text{all}}$: champions available to our team.
- Team ban set $B \subseteq C$: champions our team cannot play.
- Eligibility indicator $1_{r,c} \in \{0, 1\}$ for $r \in R, c \in C$:

$$1_{r,c} = \begin{cases} 1, & \text{if champion } c \text{ is eligible on role } r, \\ 0, & \text{otherwise.} \end{cases}$$

Utilities

- Utility matrix $u \in \mathbb{R}^{R \times C}$ with entries $u_{r,c}$ = performance/fitness score of champion c on role r.
- Larger $u_{r,c}$ means better performance.

Toy example (2 roles, 3 champions)

$$u = \begin{array}{c|ccc} & G & D & A \\ \hline \text{TOP} & 8 & 7 & 4 \\ \text{MID} & 5 & 3 & 9 \end{array}$$

Network Structure for Ban–Pick

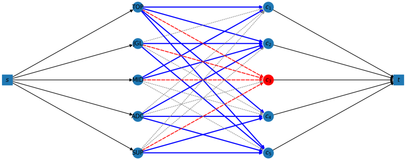
Min-cost flow viewpoint

- Node set: $V = \{s\} \cup R \cup C \cup \{t\}$.
- Arc set: $A = A_{sR} \cup A_{RC} \cup A_{Ct}$, where

$$A_{sR} = \{(s, r) : r \in R\}, \quad A_{RC} = \{(r, c) : r \in R, c \in C, 1_{r,c} = 1, c \notin B\},$$

$$A_{Ct} = \{(c, t) : c \in C\}.$$

- Capacities: $u_a = 1 \ \forall a \in A$.
- Supplies/demands: $b_s = |R|,\ b_t = -|R|,\ b_v = 0$ for $v \in V \setminus \{s, t\}$.
- Edge costs: $c_{s,r} = 0,\ c_{r,c} = -u_{r,c},\ c_{c,t} = 0$.
- One unit of flow = one pick; min-cost flow = optimal lineup.



Overall min-cost flow network (roles, champions, bans, eligibility, optimal picks).

General Min-Cost Flow (Primal Formulation)

Standard primal min-cost flow LP

- Directed graph $G = (V, A)$.
- Variables: $x_a \geq 0$ for each arc $a \in A$.
- Data:
  - Cost c_a,
  - Capacity $0 \leq x_a \leq u_a$,
  - Node supplies/demands b_v with $\sum_v b_v = 0$.

$$\begin{aligned} \min_x \quad & \sum_{a \in A} c_a x_a \\ \text{s.t.} \quad & \sum_{a \in \delta^+(v)} x_a - \sum_{a \in \delta^-(v)} x_a = b_v, && \forall v \in V, \\ & 0 \leq x_a \leq u_a, && \forall a \in A. \end{aligned}$$

Our LoL model is a structured special case of this.

LoL Ban–Pick as a Min-Cost Flow (Primal Objective)

Decision variables (flows)

- For each arc $a = (i, j) \in A$, introduce a flow variable $x_a \geq 0$.
- In particular: $x_{s,r}$ for $r \in R$, $x_{r,c}$ for $(r, c) \in A_{RC}$, and $x_{c,t}$ for $c \in C$.
- In integral optimal solutions, $x_{r,c} \in \{0, 1\}$ encodes whether role r plays champion c.

Arc costs

- $c_{s,r} = 0 \ (\forall r \in R), \quad c_{r,c} = -u_{r,c} \ (\forall (r, c) \in A_{RC}), \quad c_{c,t} = 0 \ (\forall c \in C).$

Min-cost objective

- Standard min-cost flow objective:

$$\min_x \sum_{a \in A} c_a x_a.$$

- Plugging in the LoL costs:

$$\min_x \sum_{a \in A} c_a x_a = \min_x \sum_{(r,c) \in A_{RC}} (-u_{r,c})\, x_{r,c},$$

  because arcs (s, r) and (c, t) have zero cost.
- Equivalently,

$$\max_x \sum_{(r,c) \in A_{RC}} u_{r,c}\, x_{r,c},$$

  i.e., we choose a lineup that maximizes total role–champion utility.

LoL Ban–Pick Flow Constraints

Capacity constraints

$$0 \leq x_a \leq 1 \quad \forall a \in A.$$

Explicitly,

$$0 \leq x_{s,r} \leq 1 \ (\forall r \in R), \quad 0 \leq x_{r,c} \leq 1 \ (\forall (r,c) \in A_{RC}), \quad 0 \leq x_{c,t} \leq 1 \ (\forall c \in C).$$

Flow-balance constraints

- Source s: $\sum_{r \in R} x_{s,r} = |R|$.
- Roles $r \in R$: $\sum_{c:(r,c) \in A_{RC}} x_{r,c} - x_{s,r} = 0 \quad \forall r \in R$.
- Champions $c \in C$: $x_{c,t} - \sum_{r:(r,c) \in A_{RC}} x_{r,c} = 0 \quad \forall c \in C$.
- Sink t: $\sum_{c \in C} x_{c,t} = |R|$.

Interpretation

- Exactly $|R|$ units of flow leave the source and arrive at the sink.
- With unit capacities, each role sends at most 1 unit; each champion receives at most 1 unit.
- Together this enforces "one champion per role, no duplicate champions" for the team.

LoL Ban–Pick Min-Cost Flow: Primal Summary

Decision variables

$$x_{s,r}\ (\forall r \in R), \quad x_{r,c}\ (\forall (r,c) \in A_{RC}), \quad x_{c,t}\ (\forall c \in C), \quad x_a \geq 0\ \forall a \in A.$$

Objective

$$\min_x \sum_{(r,c)\in A_{RC}} (-u_{r,c})\, x_{r,c} \quad \Longleftrightarrow \quad \max_x \sum_{(r,c)\in A_{RC}} u_{r,c}\, x_{r,c}.$$

Constraints

$$\begin{aligned}
&\text{(source)} && \sum_{r\in R} x_{s,r} = |R|, && \\
&\text{(roles)} && \sum_{c:(r,c)\in A_{RC}} x_{r,c} - x_{s,r} = 0, && \forall r \in R, \\
&\text{(champions)} && x_{c,t} - \sum_{r:(r,c)\in A_{RC}} x_{r,c} = 0, && \forall c \in C, \\
&\text{(sink)} && \sum_{c\in C} x_{c,t} = |R|, && \\
&\text{(capacity)} && 0 \leq x_{s,r} \leq 1 && \forall r \in R, \\
& && 0 \leq x_{r,c} \leq 1 && \forall (r,c) \in A_{RC}, \\
& && 0 \leq x_{c,t} \leq 1 && \forall c \in C.
\end{aligned}$$

This is a structured min-cost flow LP on a layered network (source–roles–champions–sink).

Toy Example: Optimal Lineup

Recall toy utility matrix

$$u = \begin{array}{c|ccc} & G & D & A \\ \hline \text{TOP} & 8 & 7 & 4 \\ \text{MID} & 5 & 3 & 9 \end{array}$$

Feasible lineups (no duplicates)

- (TOP G, MID A): total utility $8 + 9 = 17$.
- (TOP D, MID A): total utility $7 + 9 = 16$.
- (TOP G, MID D): total utility $8 + 3 = 11$.
- ...

Our min-cost flow model chooses

- Optimal lineup: (TOP G, MID A).
- Max utility $= 17$, min cost $= -17$.
- In the network, this corresponds to 2 units of integral flow from s to t.

Value Function & Parametric Analysis

Value function

- Let $\mathcal{F}$ denote the feasible set defined by the LoL flow-balance and capacity constraints.
- For a fixed utility matrix u, we solve

$$Z^*(u) = \min_{x \in \mathcal{F}} \sum_{(r,c) \in A_{RC}} (-u_{r,c})\, x_{r,c}.$$

- $Z^*(u)$: optimal (minimal) total cost; $-Z^*(u)$: optimal total utility.

Parametric idea

- Introduce a real-valued parameter $\theta \in \mathbb{R}$.
- Let costs depend on θ:
$$c(\theta) = c^{(0)} + \theta d.$$
- Parametric min-cost flow:
$$Z^*(\theta) = \min_{x \in \mathcal{F}} c(\theta)^\top x.$$
- For each feasible x, the cost is affine in θ; the minimum $Z^*(\theta)$ is piecewise linear (pointwise minimum of finitely many affine functions).

Parametric Buff of a Champion (with Plot)

Buffing TOP–A by θ

- Baseline utilities $u^{(0)}_{r,c}$.
- For champion A on TOP:

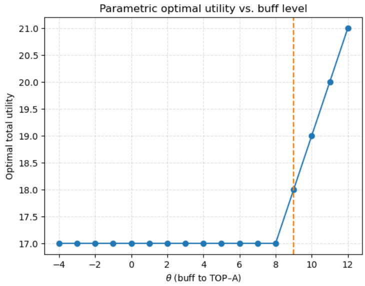
$$u_{\text{TOP},A}(\theta) = u^{(0)}_{\text{TOP},A} + \theta,$$

and $u_{r,c}(\theta) = u^{(0)}_{r,c}$ otherwise.

- Costs: $c_{r,c}(\theta) = -u_{r,c}(\theta)$ on the role–champion arcs.

Key observations from the utility plot

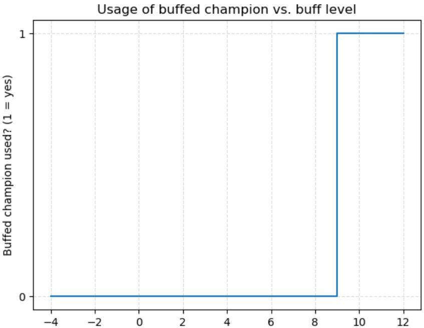
- For $\theta \le 8$: optimal total utility is constant $= 17$.
- For $\theta > 8$: optimal total utility is $9 + \theta$.
- So $-Z^*(\theta)$ is flat up to $\theta = 8$, then increases with slope 1.

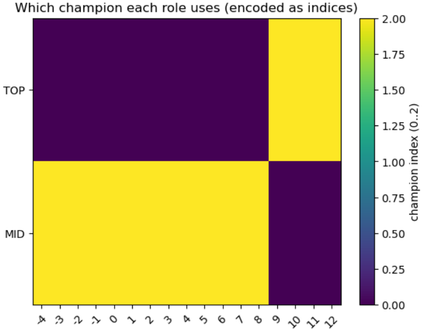


When Does TOP Start Playing the Buffed Champion?

Lineup behavior as θ changes

- For $\theta \leq 8$: optimal lineup is (TOP G, MID A) (total utility $=$ 17).
- For $\theta > 8$: optimal lineup switches to (TOP A, MID G) (total utility $= 9 + \theta$).
- The usage plot pinpoints the buff level where TOP starts playing A.
- The heatmap visualizes the role swap between G and A.





Python Implementation: Min-Cost Flow (Primal)

Reduced assignment-style formulation on role–champion arcs A_{RC}:

```
model = LPModel()
for r in R:
    for c in C:
        if eligible[r, c] and (c not in B):
            x[r, c] = model.add_var(binary=True)

# objective: min sum_{(r,c) in A_RC} (-u[r,c]) * x[r,c]
model.minimize(
    sum(-u[r, c] * x[r, c]
        for r in R for c in C
        if (r, c) in A_RC)
)

# one champion per role
for r in R:
    model.add_constr(
        sum(x[r, c] for c in C
            if (r, c) in A_RC) == 1
    )

# no duplicate champions
for c in C:
    model.add_constr(
        sum(x[r, c] for r in R
            if (r, c) in A_RC) <= 1
    )

model.solve()
```

Python Implementation: Parametric Analysis

Loop over parameter values

- Choose a grid of parameter values $\Theta = \{\theta_1, \ldots, \theta_K\}$.
- For each $\theta \in \Theta$:
  - Update utilities $u(\theta)$ (buff TOP–A by θ).
  - Re-solve the min-cost flow / assignment model.
  - Store $Z^*(\theta)$ and the optimal lineup.

```
theta_grid = [-4, -3, ..., 12]
Z_star = {}
lineup = {}

for theta in theta_grid:
    update_utilities(u, theta)  # modify u[TOP, A]
    update_objective(model, u)  # costs c_{r,c} = -u_{r,c}
    model.solve()
    Z_star[theta] = model.objective_value
    lineup[theta] = extract_lineup(x)

# Plots are generated from these recorded values.
```

Interpretation

- Parametric analysis gives a quantitative condition: TOP will swap to champion A only if the buff exceeds 8 units.
- This parametric pipeline (update utilities + re-solve the min-cost flow) extends directly to larger champion pools and more roles.

Beyond Static LP: Game-Theoretic Extension

Limitations of the current model

- Single team, single optimization problem.
- No explicit modeling of the opponent's choices.
- No sequential information: bans and picks are made in multiple rounds.

Game-theoretic / dynamic ideas

- Two players: Blue vs Red team.
- State = remaining champions + history of bans/picks.
- Each step: choose a ban or pick to maximize expected payoff.
- Could be modeled as:
  - Extensive-form (sequential) game,
  - Markov / stochastic game with value functions,
  - Stackelberg-type models (leader vs follower).
- Our current LP can be seen as the "stage payoff" inside such a dynamic model.

Summary & Thanks

Summary

- Built a primal min-cost flow model for LoL ban–pick on a layered network.
- Showed how champion utilities enter the objective via costs on role–champ arcs.
- Made the role/champion constraints explicit as flow-balance + capacity.
- Performed parametric analysis to understand the effect of buffs/nerfs.
- Implemented the model in Python and visualized piecewise-linear behavior.
- Sketched how game theory could yield a dynamic extension.

Thanks!

- Questions, comments, or wild champion balance ideas?

Hanbyul (Han) Lee – `hanbyul.lee@ucdenver.edu`