

“固定参数”其实都是你的“**假设**”。可以根据需要调整这些数值，从而观察结果如何变化

(论文中叫**灵敏度分析 Sensitivity Analysis**)。

1. 核心经济参数 (影响“山峰”的位置)

在 `simulate_season_performance` 函数中：

- **票价与客流量**: `ticket_rev = 14000 * (1 + (win_rate - 0.5)) * 60 * 20`
 - **14000**: 场馆的基础平均上座人数。如果你的球队（如印第安纳狂热）场馆更大，可以调高到 **17000**。
 - **60**: 平均票价。你可以调高（如 **80**）来测试高票价策略，或调低（如 **40**）来测试走量策略。
 - **20**: 主场比赛场数。
- **周边收入**: `merch_rev = 500000 * (1 + np.log(1 + fame_sum))`
 - **500000**: 基础周边销售额。如果有像 Caitlin Clark 这种顶级流量，可以调高到 **1,000,000**。
- **运营成本**: `profit = total_rev - salaries - 3000000`
 - **3000000 (3M)**: 球队租场馆、差旅、教练组的杂项开支。

2. 风险与胜率参数 (影响“山峰”的形状和高低)

在 `simulate_season_performance` 函数中：

- **胜率函数**: `win_rate = 1 / (1 + np.exp(-1.5 * (team_per - 5.0)))`
 - **5.0**: 联盟平均竞争力阈值。如果调高到 **6.0**，意味着联盟竞争变激烈了，同样的阵容赢球更难了。
 - **-1.5**: 敏感度。数值越大，代表“实力差一点点，胜率差很多”。
- **伤病影响**: `health_status = np.where(health_status == 0, 0.2, 1.0)`
 - **0.2**: 代表球员受伤后，该赛季只能贡献 20% 的战力。调低到 **0.0** 代表一旦受伤彻底报销。
- **海外疲劳惩罚**: `fatigue_factor = np.where(roster['plays_overseas'] == 1, 0.9, 1.0)`
 - **0.9**: 海外打球导致战力下降 10%。如果你认为海外打球影响巨大，可以调到 **0.8**。

3. 管理逻辑参数 (影响阵容的选择偏好)

- **潜力加成**: `future_bonus = young_count * 75000`
 - **75000**: 每一个年轻新秀给球队未来带来的估值增量。如果你更看重未来，可以调高到 **200,000**，这样算法会更倾向于选新人而非老将。
- **风险厌恶程度**: 在 `objective_function` 中：
 - `risk_penalty = (exp_profit - cvar_profit) * 0.5`
 - **0.5**: 风险惩罚系数。调到 **1.0** 代表老板非常胆小，算法会选出最不容易受伤、最稳的阵容；调到 **0.1** 代表老板是赌徒，只要上限高，不在乎倾家荡产。

4. 算法性能参数 (影响计算速度和“优劣”)

在 `if __name__ == "__main__":` 中：

- `generations=60`: 进化多少代。代数越多，算出来的结果越趋于“最优”，但时间更长。

- `population_size=500`: 每一代有多少种阵容组合。数值越大，搜索范围越广，越不容易陷入“局部最优”，但非常吃内存。

5. 球员库的“补丁”（影响特定球员的价格）

在 `load_and_process_data` 的 `manual_corrections` 字典里：

- 你可以手动调整任何球员的 `salary`。比如你觉得 `Caitlin Clark` 的真实商业价值 (Fame) 应该更高，或者她的薪资应该根据 2026 年新合同调高，可以直接修改这里的数字。
-

调整建议：

可以跑 3 次 代码：

1. **基准组 (Baseline)**: 使用默认参数。
2. **激进组 (Aggressive)**: 把 60 票价调到 100, `risk_penalty` 调到 0.1。
3. **稳健组 (Conservative)**: 把 `risk_penalty` 调到 1.0, `future_bonus` 调高。

然后在论文中对比这三种情况选出的球员名单有什么不同，这就是非常高阶的**对比实验分析**。