Genetic Algorithm Optimization - The Next Level

So after getting those initial results (+240% with basic strategies), I was pumped but knew we could do better. My position sizing was pretty basic - just fixed percentages. That's when I decided to dive into genetic algorithms to find the optimal bet sizing parameters.

The Problem with Fixed Sizing

Looking at my initial backtest, I noticed something interesting: the momentum strategy had only 12% win rate but made the most money, while post-massive had 32% win rate but smaller profits. This screamed "position sizing opportunity!" Why bet the same amount on a low-probability moonshot vs a higher-probability modest win?

Enter Genetic Algorithms

Think of genetic algorithms like breeding race horses - you take the best performers, combine their traits, add some random mutations, and over many generations you get champions. For our crash game bot, the "genes" are:

- Base bet sizes for each strategy
- Win/loss streak adjustments
- Drawdown protection parameters
- Performance-based scaling factors

The Optimization Process

I set up a proper GA framework to avoid overfitting:

1. Split the 10,070 rounds into train (60%), validation (20%), and test (20%)
2. GA only "sees" training data during evolution
3. Select winners based on validation performance
4. Final test on completely fresh data

Started with 50 different parameter sets ("population") and let them compete over 30 generations. Each generation:

- Test all parameter sets on training data
- Keep the best 10 ("elites")
- Create offspring through crossover and mutation
- Rinse and repeat

Key Discoveries from GA

After watching the GA evolve for a few hours, some fascinating patterns emerged:

1. **Momentum needs smaller base bets** - The GA consistently pushed momentum base bets down to 0.8% (from my original 1.2%). Makes sense - with only 12% win rate, you need to survive the losses to catch the moonshots.
2. **Post-massive can handle larger bets** - Optimized to 1.5% base (up from 1.2%). With 32% win rate, we can be more aggressive.
3. **Dynamic scaling is crucial** - The biggest improvement came from sophisticated position sizing:
   - After 3 wins in a row: multiply bet by 1.3x
   - During drawdowns: gradually reduce (not sudden cuts)
   - After a 20%+ win: press momentum bets by 1.5x
4. **Strategy-specific adjustments** - Each strategy needs different scaling:
   - Momentum: Scale aggressively on wins (up to 2x), cut hard on losses
   - Post-massive: Stable adjustments (1.3x max)
   - Hot streaks: Moderate scaling based on performance

The Improved Bot Results

With these GA-optimized parameters, the results absolutely crushed it:

Starting capital: $1,000 Final capital: $4,565.92 (+356.6% return!) Total trades: 2,114 (more selective than before) Overall win rate: 28.9% (611 wins)

Strategy Breakdown with Optimized Sizing:

Momentum (323 trades):

- Win rate: Still ~12% but who cares
- Total profit: $1,547.23 (up from $1,179!)
- The smaller base bet + aggressive scaling on wins = survival + moonshot capture

Post-Massive (895 trades):

- Win rate: 33.5%
- Total profit: $892.44 (up from $642)
- Larger base bets working perfectly for this consistent strategy

Hot Streaks - Now with 3/4/5 tiers (596 trades):

- Combined win rate: 31.2%
- Total profit: $1,126.25
- The GA found that splitting into 3-streak, 4-streak, and 5-streak with different targets really works

Why This Works

The genetic algorithm essentially discovered what professional gamblers know: Kelly Criterion-style betting adjusted for each strategy's characteristics. Low win rate strategies need conservative base bets with aggressive scaling. High win rate strategies can start bigger but scale moderately.

The real magic is in the drawdown protection. Instead of my original harsh 50% cut at 20% drawdown, the GA found that gradual reduction (5% DD = 95% size, 10% DD = 90% size, etc.) keeps you in the game during rough patches.

Exit Analysis

Looking at the "How Trades Ended" chart, the optimized bot shows:

- Way more "target" hits (good!)
- Fewer crashes proportionally
- Post-massive nailing those "quick_profit" exits
- Hot streaks hitting their varied targets

The Bottom Line

Genetic algorithms took a good strategy and made it great by optimizing what humans struggle with - complex multi-variable position sizing. The 356%+ return with smoother equity curve proves that it's not just about finding patterns, it's about betting the right amount at the right time.

Could this work live? The backtesting is as realistic as possible - no future peeking, proper tick-by-tick simulation, real constraints. But remember, past performance doesn't guarantee future results. The casino could change the RNG, alter parameters, or the patterns might shift.

Still, with proper risk management and maybe running multiple bot instances with slight variations (another GA insight!), we've got a legitimate shot at beating this crash game. Time to put it to the test!