

# The Geometry of Talent

TalentSync.Pro 0 followers

April 1, 2026

*Why Hiring Is a Network Flow Problem and What Happens When You Solve It Like One*

---

There is a peculiar kind of dysfunction that persists because it is universal. When every company struggles to hire and every candidate struggles to get hired and both sides blame the other for being unreasonable, a certain fatalism sets in. The hiring process is bad, everyone agrees — lamentable, endured. But it is bad in the way that scurvy was bad before anyone understood vitamin C. The problem is misdiagnosed. The industry has spent two decades optimizing competently at the wrong target, and the results are exactly what two decades of competent misdirected effort produces.

## The Wrong Target

Every major hiring platform, reduced to its essentials, does the same thing: an employer posts a job, candidates apply or are surfaced by a search algorithm, the employer reviews a ranked list, and the best candidates receive outreach. This is greedy per-job optimization, and it has a property that is almost never stated: candidates are a shared resource.

When Company A and Company B both want senior backend engineers, and the same three people top both of their ranked lists, one company gets lucky and the other gets nothing. Multiply this by every open role in the market and you have a system where a small number of popular candidates are deluged with attention while everyone else — candidates and companies alike — wonders what went wrong. The popular candidates are overwhelmed and often unresponsive. The less-visible candidates are underexplored. The companies that aren't the most attractive employer in their niche get scraps. The platform reports high “engagement” because so many messages were sent, which measures hiring success the way counting undelivered letters measures postal efficiency.

We tested this. At a moderate market ratio — one and a half candidates per open slot — the greedy approach delivers a hireable candidate to 24.5% of companies. Three out of four employers walk away empty-handed. This is the mathematical consequence of per-job local optimization in a globally constrained market. Better search filters, better matching scores, and better user experience all leave it intact. The allocation mechanism is the problem.

## The Right Problem

Given  $N$  candidates who each want one job, and  $M$  jobs that each want up to five candidates, assign candidates to jobs such that total match quality is maximized and every company gets someone worth talking to. This is a global optimization with capacity constraints. It has been studied since the 1950s under the name min-cost max-flow, and it runs in polynomial time.

Hiring platforms have modeled themselves as search engines — per-query operations. Matching is a market-wide operation. The difference is the difference between each person at a dinner party choosing their own seat without coordination and a seating chart that accounts for everyone's preferences simultaneously.

The formulation is clean. Build a network: a source node connects to every candidate, every candidate connects to every job they qualify for, every job connects to a sink. Candidate-to-source capacity is 1 (each person takes one job). Job-to-sink capacity is 5 (each job gets up to five introductions). The cost on each candidate-to-job arc is the negative of their compatibility score. Push flow through this network from source to sink minimizing total cost, and the emerging flow is the optimal matching.

Constraints encode directly in the network structure. A quality floor? Remove edges below a threshold. Equitable distribution so no single job hogs all the best candidates? Split each job node into a premium tier and a standard tier with a small bonus on the premium arcs. The optimizer fills one premium slot at

every job before filling second slots anywhere, because the math rewards breadth over depth. A minimum fill guarantee? Set minimum flow on job-to-sink arcs. These are structural properties of the network, respected by the solution.

There is a subtlety in the tiered-node design worth describing because it illustrates a general principle about optimization under uncertainty. The bonus on the premium tier must be calibrated relative to the scoring function's precision. If your scoring function produces scores ranging from 0.3 to 0.8 with standard deviation 0.14, a bonus of 0.10 is a gentle nudge — smaller than the typical difference between a good match and a mediocre one. The optimizer uses it to distribute quality equitably without overriding genuine quality signals. If your scoring function is imprecise — TF-IDF cosine similarity, standard deviation 0.04 — that same bonus exceeds most score differences between candidates. It overwhelms the quality signal and the optimizer routes essentially at random.

The fix is an adaptive bonus: a fixed fraction of the score spread rather than a fixed number. We use seventy percent of the standard deviation, keeping the bonus sub-noise on any scoring function. Small detail. The difference between an algorithm that works in a controlled experiment and one that works in production.

## What the Numbers Look Like

We ran this at 1,000 candidates and 200 jobs across 50 random seeds.

Greedy achieves 24.5% company satisfaction. Min-cost flow with tiered job nodes achieves 95.3%.

Every comparison is significant at  $p < 10^{-20}$ . If you ran this experiment once per second for the entire age of the universe, you would not expect to see this result by chance even once. The advantage held across the full threshold sweep from 0.40 to 0.95 hireability, and at every market ratio from 1:1 to 10:1 — growing larger as the market tightens, which is exactly when it matters most.

The real-data validation tells the same story. We took 8,000 resume-job description pairs labeled by human experts as No Fit, Potential Fit, or Good Fit. Built simulated markets from this data. Asked each algorithm to produce matches. The min-cost flow engine matched candidates to jobs that human experts would approve of 90.8% of the time. Greedy: 74.9%. Nine out of ten versus seven and a half out of ten, on data labeled by people who actually read the resumes.

The definitive test: a fifteen-week lifecycle simulation on the same real data, where introductions lead to hires, hired candidates leave the market, filled jobs close, and the algorithm keeps performing as the pool evolves. Over fifteen weeks at 1.5:1 ratio, the global optimizer produced nearly nine more hires than greedy — a 43% improvement. On a second real-world dataset of 851 resume-job pairs scored by GPT-4o, the tiered optimizer achieved 87.3% job coverage versus greedy's 40.0% at a 2:1 ratio, and 62.2% versus 20.0% at 1:1. Three independent datasets, three different labeling methodologies, the same directional result every time. The advantage is a structural property of the allocation method.

That last point bears emphasis. We tested with three completely different scoring functions — simple cosine similarity, a research-grounded multi-component scorer, and a graph-based scorer that accounts for skill prerequisites and category relationships. The algorithm's advantage over greedy ranged from sixty-four to seventy-two percentage points depending on the scorer. Greedy stayed at roughly twenty percent job coverage regardless of which scorer it used. The most sophisticated scoring function in the world, deployed in a greedy per-job architecture, leaves three out of four employers empty-handed. The bottleneck is the allocation. Improving the scoring function is sharpening the blade on a hammer.

## The Ensemble Trick

A single matching algorithm, however good, only finds what the scoring function tells it to find. If the scoring function undervalues some dimension of talent — and all scoring functions encode assumptions that are sometimes wrong — candidates who excel on that dimension are invisible to the optimizer. This is the culture-fit trap writ algorithmic: the system finds people who score well on the dimensions it measures, producing homogeneous teams that innovatively reproduce the same three ideas.

Each job gets five candidate introductions per cycle. Two are filled by the welfare-maximizing optimizer. Two are filled by the equity-distributing variant with tiered nodes. One — the important one — is filled by an exploration pass that deliberately reaches beyond the optimizer’s top picks.

The exploration slot draws from the top 30% of the viable candidate pool, weighted by the square root of inverse rank rather than by score. It picks candidates who are genuinely qualified but who are the obvious choice to nobody. Unusual skill combinations. Psychological profiles that diverge from the job’s stated ideal in ways that might indicate fresh perspective rather than poor fit. The literature calls this culture add rather than culture fit, and reports that culture-add teams produce 35% more innovative output and are 70% more likely to capture new markets.

Over a 15-week lifecycle simulation, the ensemble fills 94.2% of jobs versus 90.9% for the best pure optimization strategy. It generates slightly fewer total hires — about three fewer out of a hundred — because exploration slots convert at a lower rate. Three lost hires is the premium for market health, and it is a bargain.

The ensemble also provides something rare in hiring technology: a built-in mechanism for learning whether your own assumptions are wrong. Each slot is tagged with the strategy that produced it. If the exploration slot consistently outperforms the welfare slot for creative roles, that tells you something important about your scoring function. If it consistently underperforms for technical roles, that tells you something else. The system corrects itself through its own structure.

## Why This Matters Right Now

The algorithm’s advantage is largest in tight markets at cold start. At a 1:1 candidate-to-job ratio — the condition every new platform faces before it has built up a user base — the welfare optimizer achieves 62% market health versus greedy’s 27%. At a 10:1 ratio, when the market is saturated enough that any algorithm finds someone decent, the advantage shrinks to zero. The algorithm matters most precisely when the market is hardest.

For a platform trying to establish itself, this is the difference between retaining employers after their first matching cycle — 95% of them got a strong candidate — and losing them because 75% didn’t. Network effects in two-sided marketplaces are unforgiving. The platform that delivers value in the first cycle earns the right to build a user base.

## The Scoring Function

The algorithm is the engine; the scoring function is the steering. The default scoring function in the hiring industry — cosine similarity of skill embeddings, sometimes with seniority weighting — is roughly as informative as choosing a surgeon by how similar their resume looks to the last surgeon you hired.

In 2022, Sackett, Zhang, Berry, and Lievens published a meta-analysis that rewrote twenty-four years of received wisdom about what predicts job performance. Structured interviews now have the highest predictive validity ( $r = .58$ ), surpassing cognitive ability tests, which had held the top position since Schmidt and Hunter’s 1998 study but turned out to have been overcorrected for range restriction (corrected validity:  $r = .31$ ). Work samples and job knowledge tests followed. Unstructured interviews — the “tell me about yourself” format that dominates hiring — clocked in at  $r = .19$ . A coin flip with mild thumb-on-the-scale outperforms the unstructured interview as a predictor of performance.

We built a scoring function that takes this research seriously. Structured assessment gets the highest weight (0.28), followed by skills overlap (0.25), culture add (0.17), psychological match (0.15), seniority alignment (0.10), and role alignment (0.05). Skills overlap decomposes into coverage, depth, and surplus rather than collapsing to cosine similarity. The culture-add component explicitly rewards candidates who are aligned on core values but divergent on approach — people who share the team’s conscientiousness and stability but bring a different communication style or creative orientation.

This scoring function picks a different best candidate than cosine similarity for 72% of jobs. When we backtest against human expert labels, it picks the candidates the experts would have picked.

## The Moat Is the Data

Everything described so far — the algorithm, the scoring function, the ensemble — is reproducible. Someone could read this paper and implement the same system. What they cannot reproduce is the data that emerges from operating it as a brokered marketplace.

All communication flows through the platform. Neither party gets the other's contact information until both have confirmed intent to proceed to an offer. The platform observes every message, every response time, every non-response, every stage transition, every decline reason, every outcome. Closed-loop data on both sides of every hiring conversation.

Job boards lose visibility the moment a candidate clicks apply. Recruiters lose visibility the moment they send an introduction. ATS systems see one side of the conversation. TalentSync sees both sides, end to end, from introduction to hire or decline, with structured feedback at every stage.

This data makes the scoring function better over time, because you can calibrate compatibility scores against actual hiring outcomes rather than proxy metrics. It makes the matching algorithm better over time, because you can measure which ensemble strategy produces the best results for which market segments. And it makes gaming the system harder, because the platform can detect ghosting, bait-and-switch, and off-platform hiring.

## What Kind of Problem Is This?

Many marketplaces operate on the greedy paradigm — each participant optimizes independently, the platform aggregates results. This works when the resource being allocated is abundant. It fails when the resource is scarce and rivalrous. In that case, independent optimization produces a Nash equilibrium that is Pareto-dominated by the coordinated solution. Everyone would be better off if a coordinator allocated the resource, but no individual has the incentive or ability to coordinate unilaterally.

This is exactly the role a matching platform can play. A coordinator — one that sees the entire market, computes the globally optimal allocation, and distributes it fairly, on evidence, with a mechanism for self-correction built into the structure.

The mathematics is seventy years old. The hiring science is three years old. The combination has not existed before. And the results — 95% versus 25% company satisfaction, validated across fifty random seeds, confirmed on three independent real-world datasets, robust to scoring function choice, sustained over fifteen-week lifecycle simulations — suggest that the current state of hiring technology is a failure of imagination. The tools to solve this have existed for a lifetime. Someone just had to pick them up.