PNAS RESEARCH ARTICLE | COMPUTER SCIENCES NEUROSCIENCE

A neural algorithm for computing bipartite matchings

Sanjoy Dasgupta[a], Yaron Meirovitch[b], Xingyu Zheng[c], Inle Bush[c], Jeff W. Lichtman[b], and Saket Navlakha[c,1]

Affiliations are included on p. 9.

Edited by Christos Papadimitriou, Columbia University, New York, NY; received November 29, 2023; accepted July 5, 2024

Finding optimal bipartite matchings—e.g., matching medical students to hospitals for residency, items to buyers in an auction, or papers to reviewers for peer review—is a fundamental combinatorial optimization problem. We found a distributed algorithm for computing matchings by studying the development of the neuromuscular circuit. The neuromuscular circuit can be viewed as a bipartite graph formed between motor neurons and muscle fibers. In newborn animals, neurons and fibers are densely connected, but after development, each fiber is typically matched (i.e., connected) to exactly one neuron. We cast this synaptic pruning process as a distributed matching (or assignment) algorithm, where motor neurons "compete" with each other to "win" muscle fibers. We show that this algorithm is simple to implement, theoretically sound, and effective in practice when evaluated on real-world bipartite matching problems. Thus, insights from the development of neural circuits can inform the design of algorithms for fundamental computational problems.

neural algorithm | bipartite matching | neuromuscular circuit | circuit development | neural-inspired computing

Significance

Finding bipartite matchings is one of the oldest and most well-studied problems in computer science. This problem comes up in many guises, such as when matching donors to recipients for organ transplants, advertisers to advertising slots in marketing applications, or drivers to passengers on ride-sharing services. By studying the development of the neuromuscular circuit-a key circuit for motor control with a bipartite architecture-we found an algorithm for the bipartite matching problem. This algorithm achieves near-optimal performance, despite being fully distributed and privacy-preserving. Biologically, this algorithm demonstrates how matching architectures can scalably and robustly emerge over development despite significant variation in circuit sizes and wiring patterns. Thus, we uncovered a unique connection between computer science and neuroscience.

The development of neural circuits provides optimized network architectures for subsequent learning and behavior (1). Here, we found that the development of the neuromuscular circuit can be formulated as a distributed algorithm to form a network architecture, called a bipartite matching (Fig. 1*A*). Bipartite matchings are central to numerous optimization problems in economics, operations research, and computer science (2, 3).

The neuromuscular circuit can be viewed as a bipartite graph between N motor neurons and $M \geq N$ muscle fibers (Fig. 1*B*). To execute a particular motor command, the cortex provides all motor neurons with a common drive (4). When the drive exceeds a motor neuron's firing threshold, the neuron fires, and all of the muscle fibers it innervates contract, increasing the tension of the muscle. As the drive increases, more neurons fire and larger forces are created. The main challenge of this circuit is to efficiently generate the precise amount of force needed to perform an action.

Two properties are observed in this biological bipartite graph to address this challenge. First, the bipartite graph forms a "1-to-many matching" (Fig. 1*C*), where each motor neuron connects to one or more fibers, but each fiber is connected to exactly one neuron. In contrast, if a fiber had two inputs, then the neuron with the lower firing threshold would always drive the fiber to contract, and subsequent firing from the second neuron would provide no additional tension to the muscle, thus wasting synapses (5). A matching ensures that muscle tension is graded, where each successive motor neuron that is activated adds a discrete unit of tension; this makes muscle contraction smooth, precise, and easier to control (5, 6).

The second property is that the bipartite graph contains diverse sizes of *motor units*, defined as a motor neuron and the set of fibers it connects to. For many everyday actions, such as picking up a pencil, only a small force is required, and this force is generated by small motor units (anchored by low threshold neurons) that are recruited (activated) first. For demanding actions, such as picking up a heavy dumbbell, small motor units are first recruited, followed by larger units (anchored by high threshold neurons), which generate more force but are energetically more costly to use. This linear recruitment order of motor units, called Henneman's size principle (7, 8), has been repeatedly validated over decades of research, with few exceptions (9).

Thus, the matching, along with diversity in motor unit sizes, provides an efficient way to generate a large, discrete range of muscle forces. Nearly all mammalian muscles exhibit these properties, suggesting the importance of this motor control architecture in nature.

How is the matching specified? The connectivity between motor neurons and muscle fibers is formed during development by a competitive pruning process (10–12)

Author contributions: S.D., X.Z., and S.N. designed research; S.D., X.Z., I.B., and S.N. performed research; Y.M. and S.N. analyzed data; and S.D., Y.M., J.W.L., and S.N. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.

[1] To whom correspondence may be addressed. Email: navlakha@cshl.edu.

This article contains supporting information online at https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2321032121/-/DCSupplemental.

Published September 3, 2024.

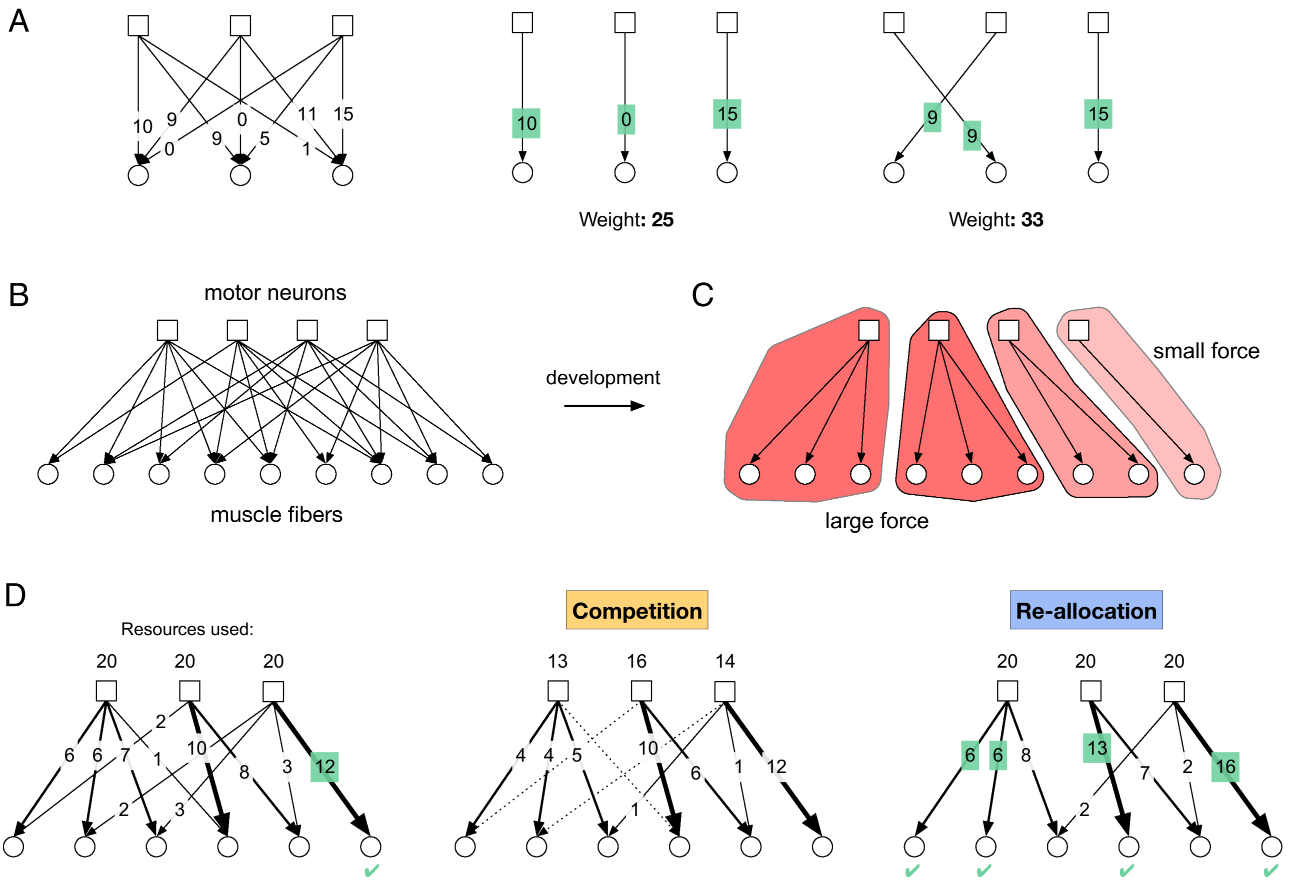


Fig. 1. Correspondence between bipartite matchings and the neuromuscular circuit. (*A*) The bipartite matching problem. The input is a weighted bipartite graph (*Left*). The desired output is a matching (i.e., a pairing-off of nodes) that has maximum total weight. The *Middle* panel shows the greedy solution, and the *Right* panel shows the optimal solution. (*B*) The neuromuscular circuit is a bipartite graph between motor neurons and muscle fibers. In newborn animals, motor neurons are densely connected to muscle fibers. (*C*) After developmental pruning, each neuron connects to at least one fiber, and each muscle fiber is matched to exactly one neuron, forming a "1-to-many" matching. Each blob corresponds to a motor unit. Small motor units are highly active, are recruited first during muscle contraction (because their neurons have low firing thresholds), and provide a small force. Large motor units are less active, are recruited last (since their neurons have large firing thresholds), and provide stronger forces. (*D*) *Left*: Example with $N = 3$ motor neurons and $M = 6$ muscle fibers. Each neuron has the same amount of total resources ($R = 20$), which are distributed among its connections. The thickness of the arrow indicates the amount of resources devoted to the connection. *Middle*: Competition occurs among all neurons connected to the same fiber, resulting in each neuron withdrawing some resources (i.e., losing some synaptic area) at that fiber, and causing some connections to be pruned (dashed edges). *Right*: The retracted resources are then reallocated to the remaining fibers the neuron still occupies.

(Fig. 1 *B* and *C*). In newborns, motor neurons and muscle fibers are densely connected; e.g., roughly, 50% of all possible connections exist in the newborn mouse interscutularis muscle (12). By adulthood, however, typically all but one synapse onto each muscle fiber is pruned. Neurons battle for fibers by allocating resources that are required to support connections and eliminate competitors; technically, these resources are acetylcholine neurotransmitters, and the competition is over acetylcholine receptor sites at the neuromuscular junction of a fiber (13). During development, all neurons that innervate the same fiber engage in an activity-dependent competition; competition causes each neuron to withdraw some resources from that fiber, and these retracted resources are then believed to be redistributed over the remaining fibers that the neuron still occupies (Fig. 1*D*). Under this model, losing one fiber increases the chances of winning other fibers, and vice versa.

The diversity in motor unit sizes emerges as a consequence of diversity in activity levels of motor neurons, which within a muscle can range from 10- to 50-fold from the least to the most active neuron (12, 14, 15). Highly active neurons (i.e., those with low activation thresholds and that are activated first) tend to compose small motor units and provide a small force, whereas less active neurons compose large motor units and provide larger forces.

Our Contributions. From a computer science perspective, we view the neuromuscular circuit development process as a distributed bipartite matching algorithm. Finding optimal matchings (also called assignments) is one of the oldest and most well-studied problems in combinatorial optimization (2, 3). In this problem, a bipartite graph is provided as input, with each edge weight indicating how beneficial it would be to match the two nodes at its endpoint; the goal is to find a matching of all the nodes with maximum total weight (Fig. 1*A*). This problem comes up frequently, such as when matching donors to recipients for transplants, advertisers to keywords for online search advertisement, or in match-making scenarios, for example, assigning medical students to hospitals for residency. However, unlike prior bipartite matching algorithms (16–18), the neuromuscular matching algorithm is fully distributed, making minimal assumptions about the computational abilities of nodes, and requiring no global memory or central coordination.

In exploring this connection, we offer the following contributions:

1. We translate insights from neuromuscular circuit development into a class of distributed algorithms for finding near-optimal bipartite matchings in terms of both efficiency (total weight) and fairness (percentage of nodes matched).
2. We interpret the algorithm as a gradient descent process, and we mathematically derive its underlying loss function and its set of stable equilibrium solutions.
3. We show empirically that this algorithm performs competitively with centralized bipartite matching algorithms, and that it can generate matchings with similar biological properties as those observed experimentally.

Results

Neuromuscular Circuit Development as a Distributed Matching Algorithm. As input, we are given a weighted bipartite graph with N motor neurons and $M \geq N$ muscle fibers. Let R_i be the total amount of resources belonging to neuron i; we assume that all $R_i = R$, that is, all neurons have the same amount of resources. Let f_i be the activity level of neuron i; the f's can be viewed as a spending denomination, that is, some neurons allocate resources using \$1 bills ($f = 1$, low activity neurons) and others use \$50 bills ($f = 50$, high activity neurons). Let $A_{ij}^{(t)} \geq 0$ be the synaptic area that neuron i controls at fiber j at time t. Synaptic area is defined as the surface area of the contact region between the motor neuron axon and the muscle fiber (19). The edge weights in the graph $A^{(0)}$ indicate the initial sizes of synaptic areas established at time $t = 0$. A primary constraint[*] for all neurons i is that $\sum_j f_i A_{ij} \leq R$, or equivalently, $\sum_j A_{ij} \leq R/f_i$.

The evolution over time of the A variables (one per synapse) can be abstracted to a two-step process: loss of synaptic area due to competition from other neurons innervating the same fiber, followed by gain of synaptic area due to reallocation of retracted resources:

$$A_{ij}^{(t-1)'} = \left(A_{ij}^{(t-1)} - \alpha \sum_{k \neq i} f_k A_{kj}^{(t-1)} \right)_+ \quad \text{(competition)} \qquad [1]$$

$$A_{ij}^{(t)} = A_{ij}^{(t-1)'} + \beta \left(\underbrace{\left(R/f_i - \sum_{j'} A_{ij'}^{(t-1)'} \right)}_{\text{retracted resources}} \underbrace{\frac{A_{ij}^{(t-1)'}}{\sum_{j'} A_{ij'}^{(t-1)'}}}_{\text{multiplicative}} \right) \text{(reallocation)} \qquad [2]$$

Eq. **1** says that neuron i loses synaptic area at fiber j based on competition from all other neurons $k \neq i$ that also synapse onto fiber j. Technically, when a neuron fires, it releases punishment signals at the fibers it contacts, which destabilizes neurotransmitter release sites from other neurons at those same fibers; neurons also release protection signals that prevent their own release sites from being lost (5, 20). The amount of punishment signal received by neuron i from each competing neuron k is proportional to the area occupied by the competing neuron (A_{kj}), as well as its activity level (f_k) (14). These punishment signals cause neuron i to retract area (resources) from fiber j. If the effect of competition renders $A_{ij}^{(t-1)'} = 0$, the synapse is pruned, and neuron i is no longer involved in the competition at fiber j. The parameter $\alpha < 1$ is a competition coefficient.

Eq. **2** says that resources retracted in the first step are reallocated over the remaining fibers the neuron still forms synaptic contacts with (21, 22). Multiplicative reallocation is consistent with the notion that large synapses have large axonal branches with more microtubules along which neurotransmitter vesicles can be transported; thus, large synapses get more resources than small synapses (22, 23). An alternative and equally simple rule to implement is constant reallocation, where each remaining synapse gets boosted by the same amount:

$$A_{ij}^{(t)} = A_{ij}^{(t-1)'} + \beta \left(\underbrace{\left(R/f_i - \sum_{j'} A_{ij'}^{(t-1)'} \right)}_{\text{retracted resources}} \underbrace{\frac{1}{\sum_{j', A_{ij'}^{(t-1)'} > 0} 1}}_{\text{constant}} \right), \text{(reallocation)} \qquad [3]$$

where the new denominator is the degree of neuron i. For both rules, we fix $\beta = 1$, which means that neurons always allocate all of their resources.

These equations follow the dynamics described in the classical models of Willshaw (24) and Barber and Lichtman (14), as well as other highly parameterized, differential equations models that incorporate varying degrees of biological realism (25–27).

To summarize, there are two factors that contribute to how motor neurons compete to "win" muscle fibers (28). First, for each fiber, there is a competition for synaptic area among all motor neurons innervating the fiber. The ability of a neuron to eliminate competing neurons is based on the size (area) it occupies at the fiber, and its activity level. Second, resources lost due to competition are redistributed over the neuron's remaining connections, increasing the chances of winning other competitions, while keeping the overall amount of resources used constant.

Relationship to Prior Matching Algorithms. In computer science, there are two broad classes of algorithms for solving the bipartite matching problem. The first use network flow techniques (29–32), and the second, which are closer to the mark, are called auction algorithms (16, 33), which produce matchings by mimicking an auction, where "buyers" (motor neurons) compete for "items" (muscle fibers) based on individual valuations. Both of these classes of algorithms, however, require centralized computation or coordination. For example, network flow techniques require computing augmenting paths or having shared memory [e.g., the parallel random-access machine model (34)], and auction algorithms require maintaining shared, globally accessible information, such as a queue of which buyers remain unmatched, and a list of the current prices of each item.

Degree diversity is considered in what are called b-matchings (38). The b vector (one entry per node in the bipartite graph) denotes the maximum number of edges that can be incident to that node in the final matching. In our case, the b-value for a fiber would be 1, and for neurons, it would be between a lower bound of 1 and some upper bound capacity (39). Both exact and approximate b-matching algorithms, however, also require global computations or shared memory (37, 40–44).

The distributed computing community has developed matching algorithms with provable approximation bounds, under models where signals (messages) are only passed locally via

---

[*] When implemented biologically, these hard constraints could be bent—e.g., a neuron may fire without releasing neurotransmitter—however, such instances do not change the effective behavior of the model.

neighbor communication. These algorithms, however, require computational liberties that are not likely biologically plausible, including unique identifiers for nodes (35–37) (i.e., messages received by a postsynaptic neuron are tagged with the identity of the presynaptic neuron who sent the message), message sizes that depend on the number of nodes in the graph (37), or invocation of complex subroutines in the matching process, such as calculating a maximal independent set (36).

Finally, the economics literature has considered problems in which there are N agents and M indivisible goods, and the goal is to assign goods to agents. Generally, each agent i has a valuation function $v_i : 2^{[M]} \rightarrow \mathbb{R}$ that captures the benefit of any *bundle* of goods. Two broad goals are *efficiency*, which corresponds to assignments with large total benefit, and *fairness*, which means roughly that assignments should not favor some agents too heavily over others (45–47). In most settings, it is computationally hard to achieve guarantees of both efficiency and fairness (48). One common assignment scheme is *round-robin*, in which a permutation of the agents is chosen and then assignment is done in a sequence of rounds in which the agents go in order and each picks one of the remaining goods (49). This scheme, however, requires a global picking order to be selected and shared.

In contrast with the above, the neural matching algorithm requires no shared memory nor centralized computation; nodes (neurons) act as individual "agents" and communication with other nodes only occurs locally at each connection point (neuromuscular junction at a fiber) (11). For the competition step, each node receives an aggregate punishment signal (a scalar value) at each connection it occupies, but it does not even "know" which nodes nor how many nodes sent these signals. For the reallocation step, each node simply redistributes its retracted resources over its own connections. Thus, the neural matching algorithm is distributed and privacy-preserving—which is important in medical (50), security (51, 52), and financial applications of matchings—since all messages are local, and nodes have no unique identifiers.

Theoretical Analysis. Does iterative application of the two update equations (competition and reallocation) converge to a matching? We analyzed the behavior of the neural matching algorithm on bipartite graphs with N neurons and M fibers. At any time, the bipartite graph is given by a matrix A that satisfies the constraints $A_{ij} \geq 0$ and $f_i \sum_j A_{ij} \leq R$, for all neurons i. Once an initial condition $A^{(0)}$ is specified, the matrix evolves deterministically according to the update rule (Eqs. **1** and **2**). At any time t, when the matrix is $A^{(t)}$, we think of the underlying connections $E^{(t)}$ in the bipartite graph as being the (neuron, fiber) pairs (i, j) for which $A_{ij}^{(t)} > 0$. As the algorithm progresses, some connections may be destroyed due to competition but new connections are never created.

Computationally, the goal of the bipartite matching problem is to find a valid assignment of maximum weight (Fig. 1*A*), while leaving no neurons nor fibers disconnected. In a valid assignment, each fiber is connected to one neuron. We define the weight of the assignment encoded by $A^{(t)}$ as $\sum_{(i,j)\in E^{(t)}} A_{ij}^{(0)}$, where $E^{(t)}$ denotes the underlying connections in $A^{(t)}$, and $A^{(0)}$ is the input matrix.

We theoretically characterized the set of equilibrium solutions: the matrices A that are left unchanged by the update rule (Theorem 1). Of particular interest are stable equilibria, those to which the system returns after small perturbations. The set of stable equilibria has a simple characterization (Theorem 2): The underlying connection graph is a union of connected components, where each component is a single neuron connected to several fibers, or a single fiber connected to several neurons. The latter category is undesirable from the point of the view of assignment, and empirically seems to occur infrequently.

We also found that the update rule can be interpreted as projected gradient descent on a loss function, whose minima correspond exactly to valid assignments. The competitive update (Eq. **1**) is the gradient step, while reallocation (Eq. **2**) is an approximation to the projection step. Finally, we showed that as long as the α parameter is not too large, no neuron nor fiber ever becomes disconnected (Theorem 3).

Thus, the neural matching algorithm typically converges to a valid and connected matching using a local, distributed gradient descent process. See *SI Appendix* for full details and proofs.

Empirical Analysis on Random and Real-World Datasets. How well does the neural matching algorithm work in practice, and does it find near-maximum weight assignments? We constructed fully connected $N \times M$ bipartite graphs with $N = 500$ and $M \in \{550, 1{,}000, 2{,}000, 5{,}000\}$. To explore a range of competitive scenarios, neuron-to-fiber synaptic weights $A_{ij}^{(0)}$ were drawn independently from three distributions (log-normal, Poisson, uniform); the former two are commonly used to model synaptic weight distributions in the brain (53). For each neuron i, we normalized the sum of its synaptic weights such that $\sum_j A_{ij}^{(0)} = R$. Starting from this initial graph $A^{(0)}$, we ran the neural matching algorithm (Eqs. **1** and **2**) for t steps. We fixed $\alpha = 0.001$ and the activity level $f_i = 1$ for each neuron i (unless otherwise noted).

The quality of the matching was judged according to two measures. *Efficiency* is the weight of the matching found; i.e., the sum of the edge weights in the input graph $A^{(0)}$ for all of the valid matched edges in the final graph $A^{(t)}$. *Fairness* is the percentage of neurons that were matched to at least one fiber. Efficiency describes the total benefit of the matching, and fairness measures equity, which is important biologically since leaving motor neurons disconnected is wasteful.

We compared the neural matching algorithm (*SI Appendix*, Algorithm 1) to four alternative algorithms: Max-Eff finds the maximum-weight matching, with no regard to fairness. This is a trivial algorithm where each fiber simply keeps its maximum weight input in $A^{(0)}$. This is an extreme upper bound on efficiency that is not used in practice because it tends to have poor fairness. Optimal (OPT) (54, 55) is a centralized algorithm that is most commonly used to balance the weight of the matching (efficiency) with the number of disconnected neurons (fairness). Greedy is a distributed algorithm where synapses are deleted stochastically with probability proportional to their inverse weight, until a single input remains for each fiber. Random is baseline weight-agnostic matching. See *Methods* for details.

Strikingly, the neural algorithm found matchings that were very close to optimal according to both quality measures (Table 1). Across all nine bipartite graphs with different sizes and weight distributions, the neural matching was at least 96% as efficient as Max-Eff. For example, with $N = 500$, $M = 5{,}000$, and log-normal initial weights, the neural algorithm found a matching whose efficiency was 99.1% of Max-Eff, compared to 63.4% for Greedy, and 7.3% for Random. Moreover, the

Table 1. Matching performance on random bipartite graphs

				Efficiency					Fairness			
Matching	N	M	Weights	Max-Eff	OPT	Neural	Greedy	Rand.	Max-Eff	OPT	Neural	Greedy
1-to-many	500	5,000	Log-normal	100.0	99.1	99.1	63.4	7.3	100.0	100.0	100.0	100.0
	500	2,000		100.0	97.8	97.9	63.1	7.5	98.5	100.0	100.0	98.0
	500	1,000		100.0	95.6	96.0	63.1	7.5	88.7	100.0	99.9	87.3
	500	550		100.0	99.1	91.9	63.4	7.3	69.1	73.6	99.8	68.0
	500	5,000	Poisson	100.0	99.5	99.6	65.5	30.6	99.9	100.0	100.0	100.0
	500	2,000		100.0	98.9	99.2	65.2	30.6	96.4	100.0	100.0	98.5
	500	1,000		100.0	97.7	98.3	65.0	30.5	83.8	100.0	100.0	85.8
	500	550		100.0	99.4	95.9	64.8	30.4	66.1	72.2	99.3	66.5
	500	5,000	Uniform	100.0	99.3	99.3	85.9	49.7	70.7	100.0	100.0	100.0
	500	2,000		100.0	98.7	98.7	85.5	49.5	49.4	100.0	100.0	98.4
	500	1,000		100.0	97.8	97.9	84.9	48.9	35.3	100.0	100.0	87.1
	500	550		100.0	98.5	97.0	83.9	49.0	26.6	58.2	99.5	66.5
1-to-1	1,000	1,000	Log-normal	—	100.0	98.6	68.1	6.8	—	100.0	97.5	63.6
	500	500		—	100.0	98.7	70.6	8.6	—	100.0	97.5	64.6
	250	250		—	100.0	98.7	69.6	10.3	—	100.0	97.3	64.9
	1,000	1,000	Poisson	—	100.0	97.3	66.6	29.5	—	100.0	96.8	63.6
	500	500		—	100.0	97.4	68.1	31.7	—	100.0	96.8	63.6
	250	250		—	100.0	97.5	69.4	34.0	—	100.0	96.9	62.3
	1,000	1,000	Uniform	—	100.0	96.6	87.8	49.7	—	100.0	96.6	62.8
	500	500		—	100.0	96.6	86.8	50.1	—	100.0	96.5	62.6
	250	250		—	100.0	97.0	85.7	52.2	—	100.0	97.0	63.2

We constructed fully connected $N \times M$ bipartite graphs, with random initial weights drawn from the indicated distribution. We compared the efficiency and fairness of the neural algorithm to four algorithms: Max-Eff (weight-maximizing upper bound), OPT (centralized algorithm that balances efficiency and fairness), Greedy (stochastic, distributed greedy algorithm), and Rand (random matching). Efficiency is normalized to Max-Eff for 1-to-many matchings, and to OPT for 1-to-1 matchings. Overall, the neural algorithm finds matchings that near-perfectly optimize both efficiency and fairness.

neural algorithm was nearly 100% fair across all nine graphs, whereas Max-Eff generally performed poorly, especially for smaller ratios of M/N (e.g., 50.6% of neurons disconnected with $N = 500, M = 2{,}000$, and uniform initial weights). Compared to OPT, the neural algorithm performed almost identically despite being decentralized.

Finally, in instances where M is only slightly larger than N, the neural algorithm achieves arguably a better balance between efficiency and fairness compared to other algorithms. For example, when $N = 500$ and $M = 550$ using Poisson weights, the neural algorithm achieves 96% efficiency and 99% fairness, whereas MaxEff and OPT achieve >99% efficiency but only roughly 70% fairness (Table 1).

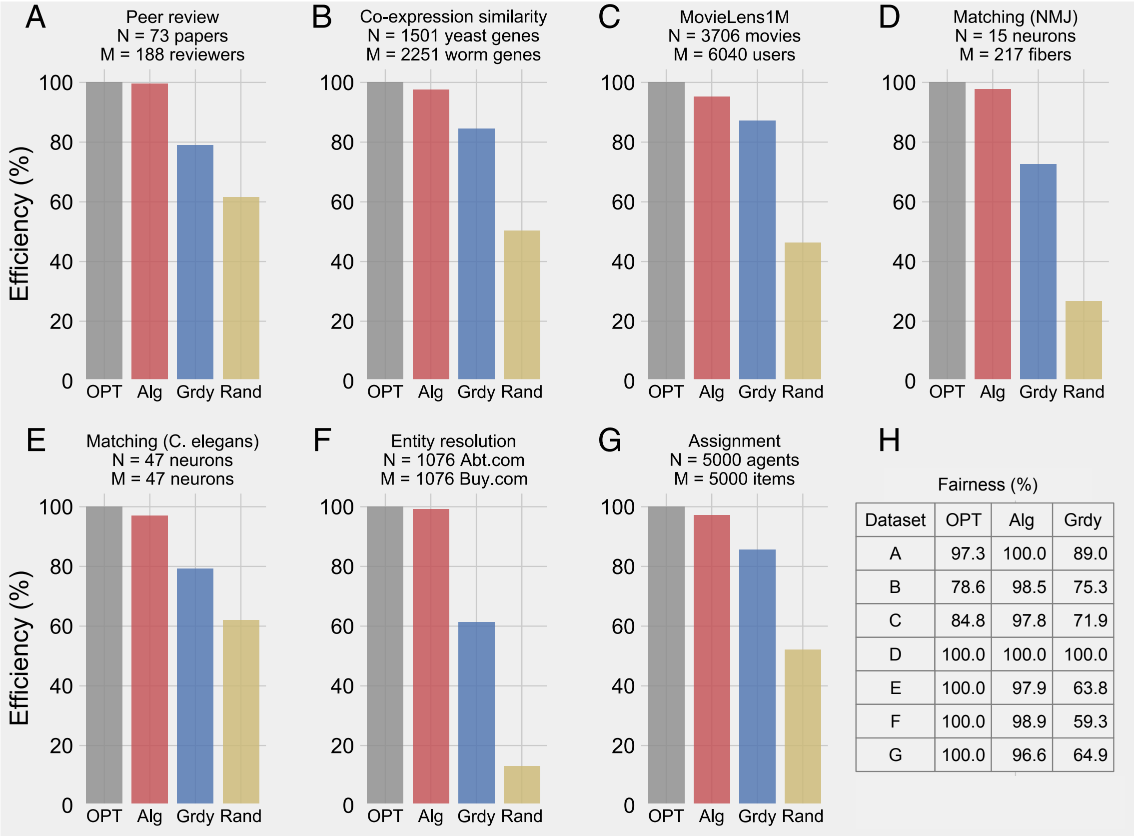
We next applied the algorithm to three real-world matching problems:

- A conference peer reviewer dataset (56): $N = 73$ papers, $M = 188$ reviewers. The weights $A_{ij}^{(0)}$ are the cosine similarity between keywords of paper *i* and reviewer *j*, indicating the relevance between the reviewer's expertise and the topics covered by the paper. The goal is to match each reviewer to a paper that maximizes the overall relevance, while covering all papers.
- A cross-species gene regulation dataset (57): $N = 1{,}501$ yeast genes, $M = 2{,}251$ roundworm genes. The genes were selected such that each yeast gene has ≥ 1 sequence-derived orthologous roundworm gene, and each roundworm gene has exactly one orthologous yeast gene. The weights $A_{ij}^{(0)}$ indicate the similarity in coexpression between genes *i* and *j* across conditions (57). The goal is to find a maximum-weight matching between the two species, which can be used to determine how overlapping transcriptional conservation is with the known sequence-based conservation.
- A movie recommendation dataset (58): $N = 3{,}706$ movies, $M = 6{,}040$ users. The weights $A_{ij}^{(0)}$ are the rating that user *j* assigns to movie *i*. The goal is to recommend one movie to each user that maximizes the overall rating score, while covering all movies.

For each dataset, the neural algorithm found a matching that was at least 95% as efficient as OPT (Fig. 2 *A*–*C*), while having better fairness (Fig. 2*H*).

Finally, we applied the neural algorithm to a bipartite graph simulating the neuromuscular wiring diagram of the mouse interscutularis muscle (a small muscle at the base of the ear) (12, 23). Here, $N = 15$ motor neurons and $M = 217$ muscle fibers in the newborn animal (12). The synaptic weights $A_{ij}^{(0)}$ were drawn from a log-normal distribution (22, 53). The N motor neurons were assigned integer activity levels f ranging from 1 to 15, mimicking a roughly 20-fold activity range from the most to the least active neuron in this muscle (14). We found that the neural algorithm matched each fiber to exactly one neuron, left no neurons disconnected, and achieved a 97.3% efficiency (Fig. 2*D*).

The process of forming the matching on the neuromuscular connectome demonstrated four biological features found in real muscle development. First, the final degree distribution of motor neurons (i.e., the number of fibers each neuron connects to) closely matched the degree distribution from the recently reconstructed connectome of the adult interscutularis muscle (12), with similar minimum and maximum motor unit sizes (Fig. 3*A*). Second, there was an inverse relationship between the activity level of a neuron (f_i) and its degree in the final matching



Fairness (%)

Dataset	OPT	Alg	Grdy
A	97.3	100.0	89.0
B	78.6	98.5	75.3
C	84.8	97.8	71.9
D	100.0	100.0	100.0
E	100.0	97.9	63.8
F	100.0	98.9	59.3
G	100.0	96.6	64.9

Fig. 2. Matching performance on real-world bipartite graphs. (*A*–*G*) Each panel denotes a dataset, with values of *N* and *M* shown on *Top*. The *x*-axis shows four matching algorithms: OPT (optimal, centralized algorithm), Alg (neural matching algorithm), Grdy (stochastic greedy algorithm), and Rand (random matching). The *y*-axis shows the efficiency (weight of the matching), normalized to OPT; higher is better. (*H*) Fairness (i.e., the number of disconnected neurons) for each algorithm on each dataset in panels (*A*–*G*). Overall, despite being decentralized, the neural algorithm finds matchings that are nearly as efficient as OPT, with similar or better fairness on average.

(Fig. 3*B*). This relationship has been dubbed a paradox (59–63), since in typical activity-dependent competition models, more activity tends to garner more connections, not fewer. In this case, however, highly active neurons compete more aggressively at the fibers they occupy, which leads to them winning fibers faster, albeit winning fewer fibers overall ("early and few") (62, 63), whereas less active neurons win "late and many" (28). Third, fibers demonstrated flip-flop behavior (Fig. 3*C*), where the largest fiber input at some time t was not always the winning input. Experimentally, if a fiber has two inputs, then the stronger of the two inputs often wins the fiber (21, 24, 64), however, flip-flops do, and are expected to, occur (11, 65)—otherwise, the initially most dominant input (in $A^{(0)}$) would always win the fiber, which, as we showed with the Max-Eff algorithm, could lead to disconnected neurons. Fourth, synapses were pruned at an exponential rate over time (Fig. 3*D*); i.e., pruning was more aggressive early in development, when many axons compete at a fiber, and then tapered off as the few remaining axons compete more slowly (66); this behavior is consistent with rates found experimentally (22).

Thus, the neural algorithm finds near-optimal matchings on random and real-world datasets, while also recapitulating key features of neuromuscular development.

Special Case of 1-to-1 Matchings. If $N = M$ and $f = 1$ for all neurons, then the problem reduces to classic 1-to-1 weighted bipartite matching. Here, each fiber again only receives one input, but now each neuron must connect to exactly one fiber. Biologically, some muscles nearly solve the 1-to-1 matching problem, such as the extraocular muscle (for eye movement control), where $M \approx 1.5N$; i.e., each neuron matches with only 1 to 2 fibers on average, and each fiber still has only 1 input (67).

As above, we first evaluated performance on random bipartite graphs with $N = M \in \{250, 500, 1{,}000\}$, and with the same breadth of weight distributions. For the neural matching algorithm, we used Eq. **3** (constant reallocation), which was slower to converge compared to multiplicative reallocation but performed slightly better. For 1-to-1 matchings, OPT is centralized and guaranteed to find the maximum weight matching with perfect fairness (*Methods*).

Across all random graphs, the neural algorithm found matchings that were at least 96% of OPT with respect to both efficiency and fairness (Table 1). The small gap in performance is mostly attributed to triangles (2 neurons → 1 fiber, or 1 neuron → 2 fibers) that form in the final matching, which represent stable equilibria of the update equations. For example, the first type of triangle forms when two neurons lose all other fibers except for one, dump all their resources into that fiber, and reach "stalemate." To resolve these triangles, we picked one random edge from the triangle, and the remaining node was left unassigned. These triangle issues may be further avoided by using a slightly modified version of the algorithm, where pruned synapses can be resurrected (*SI Appendix*), though we do not implement this version here because it is likely not biologically plausible.

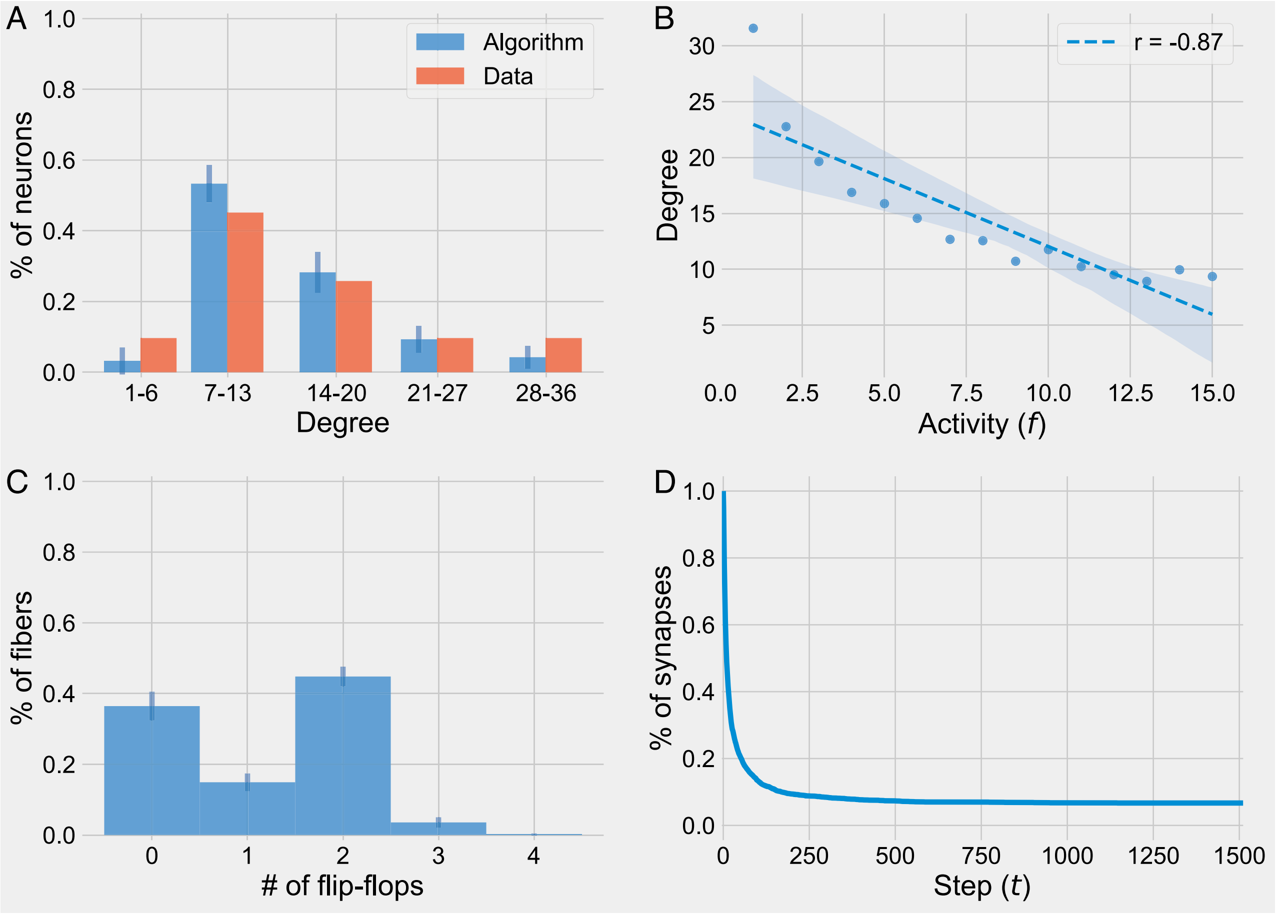


Fig. 3. Recapitulating features of the neuromuscular circuit. (*A*) Histograms comparing the neuron degrees from the neural algorithm (blue) and the actual degrees from the reconstructed mouse interscutularis muscle (12) (red). Error bars indicate SD over 10 random synaptic weight initializations. (*B*) Inverse relationship between neural activity (f, x-axis) and the final degree of the neuron (y-axis). (*C*) Histogram of fiber flip-flops: A flip-flop occurs when the largest-weight neuronal input to a fiber at time t changes at time $t + 1$. The x-axis shows the number of flip-flops that occur, and the y-axis shows the percentage of fibers. (*D*) Exponential rate of pruning during development: The x-axis shows the step number of the algorithm, and the y-axis shows the percentage of synapses remaining in the bipartite graph.

Last, we applied the neural matching algorithm to three real-world 1-to-1 matching datasets:

- A *Caenorhabditis elegans* matching dataset (68): $N = 47$ neurons in animal 1, $M = 47$ neurons in animal 2. The weights $A_{ij}^{(0)}$ are the normalized mutual information between neurons i and j, indicating how similar the activity of the two neurons are in response to odor stimuli. The goal is to find a correspondence between the same neurons in the two animals, to test the similarity of neural representations across animals (69).
- An entity resolution dataset (70): $N = 1{,}076$ Abt.com sellers, $M = 1{,}076$ Buy.com sellers. The weights $A_{ij}^{(0)}$ are the cosine similarity between product profiles of sellers, indicating how likely they correspond to the same entity (seller). The goal is to find a correspondence between the same seller on the two websites.
- An operations research assignment problem (71): $N = 5{,}000$ agents, $M = 5{,}000$ items. The weights $A_{ij}^{(0)}$ are the benefit when allocating agent i to item j. The goal is to find an assignment that maximizes the overall benefit.

Overall, we observed similar trends (Fig. 2 *E*–*H*): The neural algorithm found a matching that was on average 97 to 98% of the optimal efficiency and optimal fairness. Thus, near-optimal 1-to-1 matchings can be found with the exact same algorithm.

Discussion

In summary, we found a bipartite matching algorithm that produces near-maximum weight matchings and near-perfect fairness, despite being distributed and privacy preserving. The algorithm was inspired by circuit-level insight of the development of the neuromuscular circuit, a key neural network architecture used for motor control that is found within many vertebrate muscles.

Potential Lessons for Biology. An algorithmic perspective on neuromuscular development offers two potential lessons for biology. First, it suggests that the initial synaptic weights formed between neurons and fibers, even if random, strongly determine the architecture of the final matching circuit. Theoretically, we showed that the loss function of the update equations is stably minimized under any valid matching, and empirically, we found that the matching found is one that nearly maximizes the weights of the input. This is an intriguing aspect of the neural matching algorithm; it seems to produce approximately maximum assignments, even though the initial $A^{(0)}$ is never explicitly stored and might quickly be forgotten by the updates. Why might this occur, and why may it be beneficial biologically? One idea is that an approximately maximum-weight matching may be the fastest to find: Since winning inputs must have large weights to eliminate competitors, preserving connections that already have relatively large initial weights may speed development and convergence to a functional circuit. Indeed, if a neuron initially forms a very small contact with a fiber, it is unlikely that this synapse will grow large enough to dominate all other competitors, and if this does happen, it may take a long time. Another possibility is that the newborn connectome contains important inductive biases, and that an architectural trace of these biases is approximately preserved in the adult. For example, if some fibers have predefined types (72, 73), they may form stronger initial weights to neurons with the preferred activity level—e.g., slow twitch fibers connected to very active neurons—and the matching process would likely preserve these biases. Better understanding of how the initial $A^{(0)}$ constrains the space of matchings explored through the updates may help derive the approximation factor achieved by the algorithm. Second, our theoretical analysis assuages some debate about how development "needs to" sets the initial condition in the newborn neuromuscular connectome by showing that matchings are robustly and scalably produced in the adult given a wide variety of bipartite graphs as input. This is conceptually pleasing because the same algorithm can be applied across muscles with different sizes (values of N and M), different scales (ratios of M/N), and different competitive scenarios (weight distributions).

Algorithm Parameters. How do different settings of algorithm parameters affect the matching produced? The first parameter, α (competition coefficient), simply needs to be set small enough such that all of the connections of a single node (neuron or fiber) are not eliminated in a single competitive update. The second parameter, β (reallocation coefficient), we set to 1 to ensure that neurons allocate all of their resources in each step. We are not aware of a benefit to setting $\beta < 1$ in our model. However, in prior differential equations models (14) β reflects the rate at which a neuron can physically generate and move resources; this rate could be less than 1 for small time steps. The third parameter, f (neuron firing rates), we empirically showed was inversely related to the final degrees of the neurons (Fig. 3*B*). One potential connection here is to the *b*-matching problem, where each node i is provided as input an upper bound degree b_i on the number of final connections it can make, and the final matching must satisfy these degree constraints (i.e., large $f_i \rightarrow$ small b_i). The neural algorithm seems to also adhere to a lower bound degree constraint, such that fibers have exactly 1 connection, and neurons connect to at least 1 fiber. Thus, the neural algorithm may have an implicit bias for optimizing fairness, and careful selection of f could be used to approximate *b*-matchings.

Other Variants of Matching Problems. There are numerous variants on matching problems with potential neural analogs. First, in online matching problems (74) nodes in the graph arrive and depart over time and need to be matched accordingly. For example, when matching drivers to passengers on a ride-sharing service, both drivers and passengers enter and leave the network as they serve or are served. Biologically, new motor neurons are not believed to be gained in the adult, but all other changes—the loss of motor neurons, the gain and loss fibers (75), and the gain and loss of synapses (76, 77)—can reactivate the matching process. For example, if motor neurons are lost after injury or during the onset of disease, regeneration signals are sent by the disconnected fibers, which cause healthy motor units to sprout new connections, reinnervate the disconnected fibers, and reinitiate competition (78). These kinds of dynamics could inspire new online matching strategies and algorithms that can recover from network damage. Second, there are matching variants that promote alternative forms of diversity (39, 79–81), including allowing nodes to have different spending budgets and different degree capacities. Biologically, similar kinds of diversity may emerge from variation in resource availability (neurotransmitter production, related to R) and different firing patterns of motor neurons (related to f) (14, 23).

Generality in the Brain. The biological strategy of constructing networks by starting with an overabundance of connections followed by pruning has repeatedly demonstrated computational benefits, for example, in network design (82) and deep learning (83). For the neuromuscular circuit, overabundance may serve as an insurance policy, to ensure each fiber has at least one connection (5, 24); subsequent pruning then sculpts a matching in a manner that is adaptive to unpredictable changes during development [e.g., the loss of motor neurons during early cell migration (84)], and that is less costly than genetically prespecifying a matching. Finally, similar matching problems are solved during the development of other circuits, such as the thalamus (22), autonomic ganglia (85), the olfactory system (86), and the cerebellum (87). For example, in the newborn cerebellum, multiple climbing fibers innervate each Purkinje cell, but after development, only one climbing fiber remains (88, 89). Thus, finding matchings may be a general principle of computation in the brain.

Methods

We compared the neural algorithm with four alternative matching algorithms.

Optimal (OPT and Max-Eff). For 1-to-many matchings (i.e., when $M > N$), the weight-maximizing matching was found by reducing the problem to the classic 1-to-1 matching case using the standard node splitting trick. Each of the N neurons was split into some number of copies, and each copy was connected to the same set of fibers, with the same weights, as the original neuron. For example, if the number of copies was $k = 2$, then we created two copies, x_1 and x_2, for each original neuron x. Both x_1 and x_2 are connected to the same fibers as x, with the same weights. We then run the Hungarian 1-to-1 matching algorithm on this modified graph. If x_1 and x_2 are both matched to fibers, then x would also be matched to those fibers in the final matching.

To ensure that each fiber is matched, a lower bound on the number of copies is $k = \lceil M/N \rceil$. This is the value we used for OPT. For example, with $N = 3$ neurons and $M = 10$ fibers, there must be at least $k = 4$ copies of each neuron to ensure that each fiber is matched. This approach, however, can lead to disconnected neurons. For example, if $N = 3$ neurons and $M = 4$ fibers, and if $k = 2$, then the first neuron could be matched to the first two fibers, the second neuron could be matched to the next two fibers, and the third neuron would be left disconnected. In general, as k increases, there will be more disconnected neurons. Thus, there is a trade-off between maximizing the weight of the matching (i.e., efficiency) and keeping neurons connected (i.e., fairness).

An upper bound on the number of splits is $k = M$ (i.e., each neuron is copied M times). This is the value we used for Max-Eff. This could lead to the maximum number of disconnected nodes because a single neuron could, in principle, be matched to all M fibers. For example, on the peer review dataset ($N = 73$ papers, $M = 188$ reviewers), if $k = 188$ then 36% of papers would be unmatched. Under this case, each fiber is simply matched to its maximum weight input in the input graph, $A^{(0)}$.

The number of copies can be customized per neuron (e.g., in residency matching, each hospital has a "capacity" or maximum number of slots available for students). This is equivalent to the maximum weight b-matching problem, where each node i can have up to $b(i) = k$ matches. This, however, requires N parameters (one per neuron), and instead for simplicity, we picked a fixed value for k for all neurons.

For 1-to-1 matchings (i.e., when $M = N$), OPT is guaranteed to find the maximum weight matching, while leaving no nodes disconnected. This is equivalent to setting $k = 1$. Max-Eff is not a valid solution for 1-to-1 matchings because a neuron could connect to more than one fiber.

Greedy. This is a biologically plausible distributed algorithm, where synapses are deleted stochastically until a single input remains for each fiber. Starting from the input graph $A^{(0)}$, in each step, we pick a random fiber that has >1 inputs, and we delete one random input, selected with probability proportional to its inverse weight (smaller weight → higher probability of deletion). This method is guaranteed to converge to a matching (i.e., each fiber will have exactly one input), but it may leave neurons disconnected.

Random. For 1-to-many matchings, each neuron, in a random order, picks an unmatched fiber, until all fibers are picked. This requires $\lceil M/N \rceil$ rounds of picking, and picking is weight-agnostic. This is a type of round-robin scheme using a random ordering of the neurons (90), and it is guaranteed to have perfect (100%) fairness.

Algorithm Parameters. For all experiments, we fix $\alpha = 0.001$ and $\beta = 1.0$.

Dataset Preprocessing. For the Peer Review and MovieLens datasets, ϵ-noise was added to $A^{(0)}$ to ensure that no two rows (neurons) were exactly the same. This was useful to break symmetries.

Data, Materials, and Software Availability. Code and data have been deposited in Github (https://github.com/metalloids/nmj_matching) (91). All other data are included in the manuscript and/or *SI Appendix*.

ACKNOWLEDGMENTS. For funding, S.N. thanks the Pew Charitable Trusts, the National Institute on Deafness and Other Communication Disorders (NIDCD) of the NIH under award number 1R01DC017695, and the Simons Center for Quantitative Biology at Cold Spring Harbor Laboratory. For providing datasets, we thank Faez Ahmed, Jesse Gillis, Javier How, George Papadakis, and Hamsini Suresh. For providing feedback on the manuscript, we thank Samir Khuller, Ankit Patel, and Yang Shen.

Author affiliations: [a]Computer Science and Engineering Department, University of California San Diego, La Jolla, CA 92037; [b]Department of Molecular and Cell Biology and Center for Brain Science, Harvard University, Cambridge, MA 02138; and [c]Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, Cold Spring Harbor, NY 11724

1. A. M. Zador, A critique of pure learning and what artificial neural networks can learn from animal brains. *Nat. Commun.* **10**, 3770 (2019).
2. S. Dasgupta, C. H. Papadimitriou, U. V. Vazirani, *Algorithms* (McGraw-Hill Higher Education, 2006).
3. L. Lovász, M. D. Plummer, *Matching Theory* (AMS Chelsea Publishing Series, 2009).
4. C. J. De Luca, Z. Erim, Common drive of motor units in regulation of muscle force. *Trends Neurosci.* **17**, 299–305 (1994).
5. J. W. Lichtman, Synapse disassembly at the neuromuscular junction. *Sem. Dev. Biol.* **6**, 195–206 (1995).
6. W. Senn *et al*., Size principle and information theory. *Biol. Cybern.* **76**, 11–22 (1997).
7. E. Henneman, Relation between size of neurons and their susceptibility to discharge. *Science* **126**, 1345–1347 (1957).
8. T. C. Cope, M. J. Pinter, The size principle: Still working after all these years. *Physiology* **10**, 280–286 (1995).
9. N. J. Marshall *et al*., Flexible neural control of motor units. *Nat. Neurosci.* **25**, 1492–1504 (2022).
10. A. van Ooyen, Competition in the development of nerve connections: A review of models. *Network* **12**, 1–47 (2001).
11. S. G. Turney, J. W. Lichtman, Reversing the outcome of synapse elimination at developing neuromuscular junctions in vivo: Evidence for synaptic competition and its mechanism. *PLoS Biol.* **10**, e1001352 (2012).
12. Y. Meirovitch *et al*., Neuromuscular connectomes across development reveal synaptic ordering rules. bioRxiv [Preprint] (2021). https://doi.org/10.1101/2021.09.20.460480 (Accessed 13 September 2022).
13. M. A. Lanuza *et al*., Axonal competition and synapse elimination during neuromuscular junction development. *Curr. Opin. Physiol.* **4**, 25–31 (2018).
14. M. J. Barber, J. W. Lichtman, Activity-driven synapse elimination leads paradoxically to domination by inactive neurons. *J. Neurosci.* **19**, 9975–9985 (1999).
15. R. Hennig, T. Lømo, Firing patterns of motor units in normal rats. *Nature* **314**, 164–166 (1985).
16. G. Demange, D. Gale, M. Sotomayor, Multi-item auctions. *J. Polit. Econ.* **94**, 863–872 (1986).
17. D. P. Bertsekas, "A distributed algorithm for the assignment problem" in *Laboratory for Information and Decision Systems Working Paper* (MIT, 1979).
18. D. P. Bertsekas, Auction algorithms for network flow problems: A tutorial introduction. *Comput. Optim. Appl.* **1**, 7–66 (1992).
19. J. R. Sanes, J. W. Lichtman, Development of the vertebrate neuromuscular junction. *Annu. Rev. Neurosci.* **22**, 389–442 (1999).
20. E. Bloch-Gallego, Mechanisms controlling neuromuscular junction stability. *Cell Mol. Life Sci.* **72**, 1029–1043 (2015).
21. N. Kasthuri, J. W. Lichtman, The role of neuronal identity in synaptic competition. *Nature* **424**, 426–430 (2003).
22. J. C. Tapia *et al*., Pervasive synaptic branch removal in the mammalian neuromuscular system at birth. *Neuron* **74**, 816–829 (2012).
23. J. Lu, J. C. Tapia, O. L. White, J. W. Lichtman, The interscutularis muscle connectome. *PLoS Biol.* **7**, e32 (2009).
24. D. J. Willshaw, The establishment and the subsequent elimination of polyneuronal innervation of developing muscle: Theoretical considerations. *Proc. R. Soc. Lond. B Biol. Sci.* **212**, 233–252 (1981).
25. J. L. Gouze, J. M. Lasry, J. P. Changeux, Selective stabilization of muscle innervation during development: A mathematical model. *Biol. Cybern.* **46**, 207–215 (1983).
26. J. Stollberg, Synapse elimination, the size principle, and Hebbian synapses. *J. Neurobiol.* **26**, 273–282 (Feb1995).
27. N. Jeanpretre, P. G. Clarke, J. P. Gabriel, Competitive exclusion between axons dependent on a single trophic substance: A mathematical analysis. *Math. Biosci.* **135**, 23–54 (Jul 1996).
28. I. Nowik, S. Zamir, I. Segev, Losing the battle but winning the war: Game theoretic analysis of the competition between motoneurons innervating a skeletal muscle. *Front. Comput. Neurosci.* **6**, 16 (2012).
29. H. W. Kuhn, The hungarian method for the assignment problem. *Nav. Res. Logist. Q.* **2**, 83–97 (1955).
30. J. Edmonds, Paths, trees, and flowers. *Can. J. Math.* **17**, 449–467 (1965).
31. J. E. Hopcroft, R. M. Karp, "A n5/2 algorithm for maximum matchings in bipartite" in *Proceedings of the 12th Annual Symposium on Switching and Automata Theory (Swat 1971)* (SWAT '71, IEEE Computer Society, USA, 1971), pp. 122–125.
32. S. Micali, V. V. Vazirani, "An $o(\sqrt{|V|} \cdot |e|)$ algoithm for finding maximum matching in general graphs" in *Proceedings of the 21st Annual Symposium on Foundations of Computer Science* (SFCS '80, IEEE Computer Society, USA, 1980), pp. 17–27.
33. D. P. Bertsekas, "Athena scientific optimization and computation series" in *Network Optimization: Continuous and Discrete Methods* (Athena Scientific, 1998).
34. M. Karpiński, M. Karpinski, W. Rytter, "Oxford lecture series in mathematics and its applications" in *Fast Parallel Algorithms for Graph Matching Problems* (Clarendon Press, 1998).
35. M. Wattenhofer, R. Wattenhofer, "Distributed weighted matching" in *Proceedings of the he 18th International Conference on Distributed Computing (DISC)* (Springer, 2004), pp. 335–348.
36. T. Nieberg, "Local, distributed weighted matching on general and wireless topologies" in *Proceedings of the 5th International Workshop on Foundations of Mobile Computing* (DIALM-POMC '08, Association for Computing Machinery, New York, NY, USA, 2008), pp. 87–92.
37. Z. Lotker, B. Patt-Shamir, S. Pettie, Improved distributed approximate matching. *J. ACM* **62**, 5 (2015).
38. H. N. Gabow, P. Sankowski, Algorithms for weighted matching generalizations I: Bipartite graphs, b-matching, and unweighted f-factors. *SIAM J. Comput.* **50**, 440–486 (2021).
39. F. Ahmed, J. P. Dickerson, M. Fuge, "Diverse weighted bipartite b-matching" in *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (IJCAI'17, AAAI Press, 2017), pp. 35–41.
40. S. Assadi, S. C. Liu, R. E. Tarjan, "An auction algorithm for bipartite matching in streaming and massively parallel computation models" in *4th Symposium on Simplicity in Algorithms, SOSA 2021, Virtual Conference, January 11–12, 2021*, H. Viet Le, V. King, Eds. (SIAM, 2021), pp. 165–171.
41. M. Ghaffari, C. Grunau, S. Mitrović, "Massively parallel algorithms for b-matching" in *Proceedings of the 34th ACM Symposium on Parallelism in Algorithms and Architectures* (SPAA '22, Association for Computing Machinery, New York, NY, USA, 2022), pp. 35–44.
42. S. Dobzinski, N. Nisan, S. Oren, "Economic efficiency requires interaction" in *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing* (STOC '14, Association for Computing Machinery, New York, NY, USA, 2014), pp. 233–242.
43. T. E. Anderson, S. S. Owicki, J. B. Saxe, C. P. Thacker, High-speed switch scheduling for local-area networks. *ACM Trans. Comput. Syst.* **11**, 319–352 (Nov1993).
44. N. McKeown, The islip scheduling algorithm for input-queued switches. *IEEE/ACM Trans. Netw.* **7**, 188–201 (1999).
45. H. R. Varian, Equity, envy, and efficiency. *J. Econ. Theory* **9**, 63–91 (1974).
46. S. Bouveret, Y. Chevaleyre, N. Maudet, "Fair allocation of indivisible goods" in *Handbook of Computational Social Choice*, F. Brandt, V. Conitzer, U. Endriss, J. Lang, A. D. Procaccia, Eds., (Cambridge University Press, 2016).
47. B. Plaut, T. Roughgarden, Almost envy-freeness with general valuations. *SIAM J. Discrete Math.* **34**, 1039–1068 (2020).
48. H. Aziz, B. Li, H. Moulin, X. Wu, Algorithmic fair allocation of indivisible items: A survey and new questions. *ACM SIGecom Exchanges* **20**, 24–40 (2022).

49. S. Bouveret J. Lang, "A general elicitation-free protocol for allocating indivisible goods" in *Proceedings of the International Joint Conference on Artificial Intelligence* (AAAI Press, 2011).
50. M. Breuer, U. Meyer, S. Wetzel, "Privacy-preserving maximum matching on general graphs and its application to enable privacy-preserving kidney exchange" in *Proceedings of the Twelveth ACM Conference on Data and Application Security and Privacy* (ACM, 2022).
51. M. Blanton, S. Saraph, "Oblivious maximum bipartite matching size algorithm with applications to secure fingerprint identification" in *Computer Security – ESORICS 2015*, G. Pernul, P. Y. A. Ryan, E. Weippl, Eds. (Springer International Publishing, Cham, 2015), pp. 384–406.
52. D. Panayiotis, A. Triastcyn, B. Faltings, "A Distributed Differentially Private Algorithm for Resource Allocation in Unboundedly Large Settings" in *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, (International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2022), pp. 327–335.
53. G. Buzaki, K. Mizuseki, The log-dynamic brain: How skewed distributions affect network operations. *Nat. Rev. Neurosci.* **15**, 264–278 (2014).
54. F. Bourgeois, J.-C. Lassalle, An extension of the munkres algorithm for the assignment problem to rectangular matrices. *Commun. ACM* **14**, 802–804 (1971).
55. D. F. Crouse, On implementing 2D rectangular assignment algorithms. *IEEE Trans. Aerosp. Electron. Syst.* **52**, 1679–1696 (2016).
56. M. Karimzadehgan, C. Zhai, "Constrained multi-aspect expertise matching for committee review assignment" in *Proceedings of the 18th ACM Conference on Information and Knowledge Management* (Association for Computing Machinery, New York, NY, USA, 2009).
57. M. Crow, H. Suresh, J. Lee, J. Gillis, Coexpression reveals conserved gene programs that co-vary with cell type across kingdoms. *Nucleic Acids Res.* **50**, 4302–4314 (2022).
58. F. M. Harper, J. A. Konstan, The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.* **5**, 1–19 (2015).
59. R. R. Ribchester, T. Taxt, Motor unit size and synaptic competition in rat lumbrical muscles reinnervated by active and inactive motor axons. *J. Physiol.* **344**, 89–111 (1983).
60. R. M. Ridge, W. J. Betz, The effect of selective, chronic stimulation on motor unit size in developing rat muscle. *J. Neurosci.* **4**, 2614–2620 (1984).
61. R. J. Balice-Gordon, J. W. Lichtman, Long-term synapse loss induced by focal blockade of postsynaptic receptors. *Nature* **372**, 519–524 (1994).
62. E. M. Callaway, J. M. Soha, D. C. Van Essen, Competition favouring inactive over active motor neurons during synapse elimination. *Nature* **328**, 422–426 (1987).
63. E. M. Callaway, J. M. Soha, D. C. Van Essen, Differential loss of neuromuscular connections according to activity level and spinal position of neonatal rabbit soleus motor neurons. *J. Neurosci.* **9**, 1806–1824 (1989).
64. M. Buffelli *et al.*, Genetic evidence that relative synaptic efficacy biases the outcome of synaptic competition. *Nature* **424**, 430–434 (2003).
65. M. K. Walsh, J. W. Lichtman, In vivo time-lapse imaging of synaptic takeover associated with naturally occurring synapse elimination. *Neuron* **37**, 67–73 (2003).
66. D. Purves, J. W. Lichtman, Elimination of synapses in the developing nervous system. *Science* **210**, 153–157 (1980).
67. D. Purves, D. Fitzpatrick, G. J. Augustine, S. M. Williams, "The motor unit" in *NCBI Bookshelf* (Oxford University Press, ed. 2, 2001).
68. J. J. How, S. Navlakha, S. H. Chalasani, Neural network features distinguish chemosensory stimuli in *Caenorhabditis elegans*. *PLoS Comput. Biol.* **17**, e1009591 (2021).
69. E. R. Soucy, D. F. Albeanu, A. L. Fantana, V. N. Murthy, M. Meister, Precision and diversity in an odor map on the olfactory bulb. *Nat. Neurosci.* **12**, 210–220 (2009).
70. G. Papadakis, V. Efthymiou, E. Thanos, O. Hassanzadeh, "Bipartite Graph Matching Algorithms for Clean-Clean Entity Resolution: An Empirical Evaluation" in International Conference on Extending Database Technology (EDBT, 2022), vol. 2, pp. 462-474.
71. J. E. Beasley, Or-library: Distributing test problems by electronic mail. *J. Oper. Res. Soc.* **41**, 1069–1072 (1990).
72. J. V. Chakkalakal, S. Kuang, M. Buffelli, J. W. Lichtman, J. R. Sanes, Mouse transgenic lines that selectively label Type I, Type IIA, and Types IIX+B skeletal muscle fibers. *Genesis* **50**, 50–58 (2012).
73. Y. I. Lee, Differences in the constituent fiber types contribute to the intermuscular variation in the timing of the developmental synapse elimination. *Sci. Rep.* **9**, 8694 (2019).
74. V. V. Vazirani, F. Echenique, N. Immorlica, Online and Matching-Based Market Design (Cambridge University Press, 2023).
75. M. A. LaBarge, H. M. Blau, Biological progression from adult bone marrow to mononucleate muscle stem cell to multinucleate muscle fiber in response to injury. *Cell* **111**, 589–601 (2002).
76. M. M. Rich, J. W. Lichtman, In vivo visualization of pre- and postsynaptic changes during synapse elimination in reinnervated mouse muscle. *J. Neurosci.* **9**, 1781–1805 (1989).
77. T. Gordon, Peripheral nerve regeneration and muscle reinnervation. *Int. J. Mol. Sci.* **21**, 8652 (2020).
78. E. Martineau, A. Di Polo, C. Vande Velde, R. Robitaille, Dynamic neuromuscular remodeling precedes motor-unit loss in a mouse model of ALS. *eLife* **7**, 10 (2018).
79. J. P. Dickerson, K. A. Sankararaman, A. Srinivasan, P. Xu, "Balancing relevance and diversity in online bipartite matching via submodularity" in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence* (AAAI Press, 2019).
80. F. Eskandanian, B. Mobasher, "Using stable matching to optimize the balance between accuracy and diversity in recommendation" in *Proceedings of the 28th ACM Conference on User Modeling, Adaptation and Personalization, UMAP 2020*, Genoa, Italy, July 12–18, 2020, T. Kuflik, I. Torre, R. Burke, C. Gena, Eds. (ACM, 2020), pp. 71–79.
81. A. Louis, M. Nasre, P. Nimbhorkar, G. S. Sankar, "Online algorithms for matchings with proportional fairness constraints and diversity constraints" in *European Conference on Artificial Intelligence*, (IOS Press, 2023), pp. 1601–1608.
82. S. Navlakha, Z. Bar-Joseph, A. L. Barth, Network design and the brain. *Trends Cogn. Sci.* **22**, 64–78 (2018).
83. D. Blalock, J. J. Ortiz Gonzalez, J. Frankle, J. Guttag, "What is the state of neural network pruning?" in *Proceedings of machine learning and systems*, (2020), pp. 129–146.
84. R. W. Oppenheim, Cell death during development of the nervous system. *Annu. Rev. Neurosci.* **14**, 453–501 (1991).
85. D. Purves, J. W. Lichtman, Formation and maintenance of synaptic connections in autonomic ganglia. *Physiol. Rev.* **58**, 821–862 (Oct1978).
86. S. Fujimoto *et al.*, Activity-dependent local protection and lateral inhibition control synaptic competition in developing mitral cells in mice. *Dev. Cell* **58**, 1221–1236 (Jul2023).
87. A. M. Wilson *et al.*, Developmental rewiring between cerebellar climbing fibers and purkinje cells begins with positive feedback synapse addition. *Cell Rep.* **29**, 2849–2861 (Nov2019).
88. M. Kano, T. Watanabe, N. Uesaka, M. Watanabe, Multiple phases of climbing fiber synapse elimination in the developing cerebellum. *Cerebellum* **17**, 722–734 (Dec2018).
89. K. Hashimoto, M. Kano, Synapse elimination in the developing cerebellum. *Cell Mol. Life Sci.* **70**, 4667–4680 (Dec2013).
90. A. Bogomolnaia, H. Moulin, A new solution to the random assignment problem. *J. Econ. Theory* **100**, 295–328 (2001).
91. S. Navlakha, Data and code for "A neural algorithm for computing bipartite matchings". GitHub. https://github.com/metalloids/nmj_matching. Deposited 25 April 2024.