Physicist's view on the unbalanced k-cardinality assignment problem

Patrice Koehl [1] and Henri Orland [2]
[1]*Department of Computer Science University of California, Davis, California 95616, USA*
[2]*Université Paris-Saclay, CNRS, CEA, Institut de Physique Théorique, 91191 Gif-sur-Yvette, France*

(Received 8 August 2023; revised 29 April 2024; accepted 13 June 2024; published 2 July 2024)

The k-cardinality unbalanced assignment problem asks for assigning k "agents" to k "tasks" on a one-to-one basis, while minimizing the total cost associated with the assignment, with the total number of agents N and the total number of tasks M possibly different and larger than k. While many exact algorithms have been proposed to find such an optimal assignment, these methods are computationally prohibitive when the problem is large. We propose an approach to solving the k-cardinality assignment problem using techniques adapted from statistical physics. This paper provides a full description of this formalism, including all the proofs of its main claims. We derive a strongly concave free-energy function that captures the constraints of the k-assignment problem at a finite temperature. We prove that this free energy decreases monotonically as a function of β, the inverse of temperature, to the optimal assignment cost, providing a robust framework for temperature annealing. We also prove that for large enough β values the exact solution to the k-assignment problem can be derived using simple round-off to the nearest integer of the elements of the computed assignment matrix. We show that this framework can be adapted to handle degenerate k-assignment problems. We describe a computer implementation of our framework that is optimized for the GPU parallel architecture, using the library CUDA. This implementation is found to be as efficient as state-of-the-art implementations of parallel Hungarian algorithms on generic assignment problems, and orders of magnitude faster than those algorithms for pathological assignment cases.

DOI: 10.1103/PhysRevE.110.014108

I. INTRODUCTION

Assigning a certain number of "tasks" to be performed by "agents," with each agent-task pairing being associated with a cost, is a fundamental problem in combinatorial optimization referred to appropriately as the "assignment problem." What is understood as tasks and agents is problem specific. In most cases, the solution corresponds to the assignment with the minimal total cost, i.e., the sum of the costs of the agent-task that are paired. When the numbers of tasks and agents are equal, each agent is assigned exactly one task, and all tasks have been assigned. The problem is then referred to as the balanced assignment problem, or alternatively, using the language of graph theory, as the bipartite weighted matching problem (for a comprehensive analysis of assignment problems, see, for example, Ref. [1]). In many applications, however, the numbers of tasks and agents may differ: the problem is then referred to as the unbalanced assignment problem. Interests in solving the balanced and unbalanced problems have been stimulated by applications in operational research, economics, and data sciences, among others. With such a wide range of applications, assignment problems have been and remain a topic of research of equal importance for mathematicians, statisticians, and computer scientists. In this paper, we are interested in the unbalanced k-cardinality assignment problem, in which the number of assignments is preset to an integer value k, with k smaller or equal to the number of tasks and agents. We propose an approximate solution to this problem based on mean-field theory and show that it can be modified to yield an exact solution in nondegenerate as well as in degenerate situations in a computer efficient manner.

There are many types of unbalanced assignment problems. All of them consider a few tasks that differ from the number of agents. Some problems allow for multiple tasks to be assigned to the same agent to compensate for the imbalance (when the number of tasks is bigger than the number of agents), while the most common problem still assigns a single task to one agent, leaving some tasks and/or some agents unmatched. This problem, referred to in the literature as the k-cardinality assignment problem [2] is the main topic of this paper. The generalization to unbalanced assignment problems allowing for agents performing multiple tasks or tasks performed by multiple agents will be considered in a future paper.

Let T be the set of tasks, and A the set of agents, with cardinalities $N_1 = |T|$ and $N_2 = |A|$. For sake of simplicity, we will assume $N_1 \geqslant N_2$, but all arguments below would stay the same if it was the reverse. The textbook approach for solving the corresponding unbalanced assignment problem is to reduce it to a balanced assignment problem. The simplest reduction is to add $N_1 - N_2$ ghost agents, define arbitrary costs (usually with value 0) between all the tasks and those agents, and then solve the balanced assignment problem with N_1 tasks and N_1 agents. While simple, this approach is not efficient in terms of time and space complexities, especially when the difference between N_1 and N_2 is large. In addition, it is limited to assigning a task to all agents (or assigning an agent to all tasks if $N_2 \geqslant N_1$), which limits the generality of the problem. Instead, we introduce a constant k, defined as the cardinality of the maximum matching between T and A. k is smaller than N_1 and N_2 and at most $k = \min(N_1, N_2)$ (the latter being the most common case). The problem is then to identify a subset

T_1 of T and a subset A_1 of A, with $|T_1| = |A_1| = k$, and a bijection f between T_1 and A_1 such that

$$V(k) = \sum_{i \in A_1} C(i, f(i)) \tag{1}$$

is minimal, where $C(i, f(i))$ is the cost of assigning $i \in A$ with $f(i) \in T$. There are polynomial time algorithms that directly solve this k-cardinality assignment problem without relying to this transformation to a balanced problem. For example, the Hungarian algorithm can readily be adapted to the unbalanced case, with a time complexity of $O(N_1 N_2 k + k^2 \ln(\min(N_1, N_2))$ [3]. This algorithm is fast when k is small. Even if we assume $N_1 > N_2$ and set $k = N_2$, direct application of this algorithm would be of order $N_1 N_2^2$, compared to an order N_1^3 if we had made the reduction to a balanced case, a difference that can be significant if $N_2 \ll N_1$.

The Hungarian and most other existing algorithms [2,4,5] for solving the unbalanced assignment problem are iterative methods aimed at finding the best bijection f within the discrete set of all possible partial permutations. These methods are sequential in nature and therefore not easily amenable to parallelization. There is still significant effort put into parallelizing those assignment algorithms, especially since the introduction of GPGPU (for example, see Refs. [6,7] and references therein). These efforts are associated with finding ways to parallelize parts of the algorithms, and not their actual overall flowchart. We propose instead to radically change the approach for solving unbalanced k-cardinality assignment problems using continuous systems. Our approach is motivated by statistical physics. It is a generalization of a method we have recently proposed to solve the balanced assignment problem [8]. Our goals in this paper are to

(i) Establish and validate a continuous framework for solving the unbalanced k-cardinality assignment problem using statistical physics,

(ii) Establish that, in the generic case in which the k-assignment problem has a unique solution, the framework proposed above is guaranteed to converge arbitrarily close to that solution, both in term of energy and assignment matrix,

(iii) Derive a modification of the method that is guaranteed to find at least one solution for degenerate k-cardinality assignment problems with multiple solutions, and

(iv) Demonstrate that the implementation of this framework can be efficiently parallelized on a general-purpose GPU.

We emphasize that this formalism is not a mere adaptation, but a full generalization of the framework we develop for solving assignment problems [8]. In particular, new in this paper are:

(i) A method to account for the fact that some tasks and/or some agents are unassigned. We introduce indicator functions over the sets of tasks and agents that are optimized along with the transportation plan.

(ii) Establish the proofs of validity and convergence of our algorithm for both the unbalanced case and the k-cardinality assignment problem. The main results are provided in the text, while the proof themselves are relegated to the Appendix for more clarity. For the general balanced assignment problem, those proofs relied heavily on the fact that the corresponding transportation matrices are permutation matrices that are the extreme points of the well characterized convex set of doubly stochastic matrices (according to the Birkhoff–von Neumann theorem [9,10]). For the unbalanced k-cardinality assignment problem, the transportation matrices are sub-permutation matrices with a fixed total sum of their elements (the value k that defines the number of assignments). They belong to the convex set of doubly substochastic matrices with fixed sums of their elements, with properties akin to the Birkhoff–von Neumann theorem (these properties are discussed in Appendix A). The use of those properties to derive the convergence of our algorithm for solving the k-cardinality assignment problem is new.

Finally, we note that the method we introduce in this paper is designed for unbalanced problems but remains valid for balanced cases.

The paper is organized as follows. In Sec. II and III we describe in detail the unbalanced k-cardinality assignment problem and the framework we propose to solve this problem. Proofs of all important properties of this framework are provided in the Appendices. Section IV briefly describes the implementation of the method in a C++ program, UMatching. We highlight in this section the steps that have been taken to parallelize the algorithm, as well as its adaptation to graphics processing units (GPUs). Section V covers specifically the spacial case of degenerate k-cardinality assignment problems. In Sec. VI, we present some applications, with comparison to the standard Hungarian algorithm. We conclude with a discussion on future developments in Sec. VII.

II. k-CARDINALITY UNBALANCED ASSIGNMENT PROBLEM

A. k-Cardinality assignment problem

In the Introduction, we related the k-cardinality assignment problem to assigning k tasks to k agents, among N_1 and N_2 tasks and agents, respectively. We should note, however, that the assignment problem is more general than that and that "tasks" and "agents" should be seen as placeholders. Here we provide a more general mathematical framework for the k-cardinality assignment problem.

We consider two sets of points S_1 and S_2 with cardinalities N_1 and N_2, respectively. We encode the cost of transportation between S_1 and S_2 as a positive matrix $C(i, j)$ with $(i, j) \in [1, N_1] \times [1, N_2]$. We set the number of assignments between points in S_1 and S_2 to be k, a constant, with $k \leqslant \min(N_1, N_2)$. The unbalanced k-cardinality assignment problem (or k-assignment problem in short) can then be formulated as finding a partial permutation matrix G of rank k that defines the correspondence between points in S_1 and points in S_2. This matrix is found by minimizing the matching cost U defined as

$$U(G, C) = \sum_{i,j} G(i, j) C(i, j), \tag{2}$$

where the summations extend over all i in S_1 and j in S_2. In this equation, C is given, while G is variable. The minimum of U is to be found for the values of $G(i, j)$ that satisfy the

following constraints

$$\begin{aligned}
\forall i, \quad & \sum_j G(i,j) = n_1(i), \\
\forall j, \quad & \sum_i G(i,j) = n_2(j), \\
\sum_i n_1(i) = \sum_j n_2(j) = \sum_i \sum_j G(i,j) &= k, \\
\forall (i,j), \quad G(i,j) &\in \{0,1\}, \\
\forall i, \quad n_1(i) &\in \{0,1\}, \\
\forall j, \quad n_2(j) &\in \{0,1\},
\end{aligned} \tag{3}$$

where indices i and j are associated with S_1 and S_2, respectively. In these equations, $G(i,j)$, $n_1(i)$, and $n_2(j)$ are unknown, defining a total of $N_1N_2 + N_1 + N_2$ variables, while k is a constant.

The solution to the unbalanced k-cardinality assignment problem defines the indicator functions n_1^* and n_2^* on S_1 and S_2, respectively, which identify the subsets of S_1 and S_2 that are in correspondence, the partial permutation matrix G^* that defines those correspondence, and the minimum matching cost $U^* = U(G^*, C)$.

Minimizing Eq. (2) under the constraints (3) is a discrete optimization problem, namely an integer linear program problem. The corresponding transportation matrix G is a k partial permutation matrix (see Appendix A for a brief review on such matrices and their properties). We solve it using a statistical physics approach by rephrasing it as a temperature-dependent problem with real variables, with the integer optimal solution found at the limit of zero temperature. This relaxed version of the unbalanced assignment problem is a special case of a discrete optimal transport (OT) problem [11,12]. The solutions of the relaxed problem are doubly substochastic matrices (see Appendix A). Many methods have been proposed for solving the OT problem, from directly solving the linear system to solving entropy-regularized version of this system [13]. Here we introduce a modified version of our statistical physics approach for solving this problem [14,15]. We had presented a simplified version to solve the balanced assignment problem [16]. The version that is presented below is more general. It allows for solving the unbalanced k-cardinality assignment problem as well as the balanced assignment problem, since for the balanced case, we just need to set $N_1 = N_2$ and $k = N_1$.

B. Effective free energy for the unbalanced k-cardinality assignment problem

Solving the unbalanced k-cardinality assignment problem amounts to finding the minimum of a function defined by Eq. (2) over the space of possible partial mappings between the two discrete sets of points considered. If this function is reworded as an "energy," then statistical physics allows for a different perspective on how to solve this problem. Indeed, finding the minimum of an energy function is then equivalent to finding the most probable state of the system it characterizes. In the unbalanced k-cardinality assignment problem between two sets S_1 and S_2, the "system" is identified with the different binary transportation plans between S_1 and S_2 that satisfy the constraints (3). Those plans belong to the polytope of partial permutation matrices of rank k, which we have denoted as $P_{N_1,N_2}(k)$. Each state in this system is identified with a transportation plan $G \in P_{N_1,N_2}(k)$, and its corresponding energy $U(G,C)$ is defined in Eq. (2). The probability $P(G, n_1, n_2)$ associated with a transportation plan G and indicator functions n_1 and n_2 is defined as

$$P(G, n_1, n_2) = \frac{1}{Z(\beta)} e^{-\beta U(G,C)}. \tag{4}$$

In this equation, $\beta = 1/k_B T$ where k_B is the Boltzmann constant and T the temperature, and $Z(\beta)$ is the partition function computed over all states of the system. This partition function is given by

$$\begin{aligned}
Z_\beta &= e^{-\beta \mathcal{F}_\beta} \\
&= \int_{G \in P_{N_1,N_2}(k)} dG \sum_{i=1}^{N_1} \sum_{n_1(i)\in\{0,1\}} \sum_{j=1}^{N_2} \sum_{n_2(j)\in\{0,1\}} e^{-\beta U(G,C)},
\end{aligned} \tag{5}$$

where $\mathcal{F}(\beta)$ is the free energy of the system. This free energy is of limited practical interest as it cannot be computed explicitly. We propose a scheme for approximating it using the saddle point approximation.

Taking into account the constraints in Eqs. (3) the partition function can be written as

$$\begin{aligned}
Z_\beta = & \sum_{G(i,j)\in\{0,1\}} \sum_{n_1(i)\in\{0,1\}} \sum_{n_2(j)\in\{0,1\}} e^{-\beta \sum_{i,j} C(i,j) G(i,j)} \\
& \times \prod_i \delta\left(\sum_j G(i,j) - n_1(i) \right) \\
& \times \prod_j \delta\left(\sum_i G(i,j) - n_2(j) \right) \delta\left(\sum_{i,j} G(i,j) - k \right).
\end{aligned} \tag{6}$$

The sums impose that the $G(i,j)$, $n_1(i)$, and $n_2(j)$ take values of 0 or 1 only. The constraints are imposed through the δ functions (with $\delta(x) = 1$ if $x = 0$, and $\delta(x) = 0$ otherwise). We use the Fourier representation of those δ functions, thereby introducing new auxiliary variables x, $\lambda(i)$, and $\mu(j)$, with $i \in [1, N_1]$ and $j \in [1, N_2]$. The partition function can then be written as (up to a multiplicative constant), after rearrangements

$$\begin{aligned}
Z_\beta = & \int_{-\infty}^{+\infty} \prod_i d\lambda(i) \int_{-\infty}^{+\infty} \prod_j d\mu(j) \int_{-\infty}^{+\infty} dx e^{i\beta k x} \\
& \times \sum_{n_1(i)\in\{0,1\}} \sum_{n_2(j)\in\{0,1\}} e^{\beta(\sum_i i\lambda(i) n_1(i) + \sum_j i\mu(j) n_2(j))} \\
& \times \sum_{G(i,j)\in\{0,1\}} e^{-\beta \sum_{i,j} G(i,j)(C(i,j) + i\lambda(i) + i\mu(j) + ix)},
\end{aligned} \tag{7}$$

where i is the imaginary square root of unity ($i^2 = -1$). Note that we have scaled the auxiliary variables x, $\boldsymbol{\lambda}$, and $\boldsymbol{\mu}$ by a factor β for scale consistency with the energy term. Performing the summations over the variables $G(i,j)$, $n_1(i)$, and $n_2(j)$,

we get

$$Z_\beta = \int_{-\infty}^{+\infty} \prod_i d\lambda(i) \int_{-\infty}^{+\infty} \prod_j d\mu(j) \int_{-\infty}^{+\infty} dx e^{-\beta F_\beta(\boldsymbol{\lambda},\boldsymbol{\mu},x)},$$

where $F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)$ is a functional, or effective free energy that depends on the variables $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$, and x that is defined by

$$\begin{aligned} F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x) = &-\frac{1}{\beta}\sum_i \ln[1 + e^{i\beta\lambda(i)}] \\ &-\frac{1}{\beta}\sum_j \ln[1 + e^{i\beta\mu(j)}] - ikx \\ &-\frac{1}{\beta}\sum_{i,j} \ln[1 + e^{-\beta(C(i,j)+i\lambda(i)+i\mu(j)+ix)}]. \end{aligned} \quad (8)$$

The effective free energy $F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)$ depends on $N_1 + N_2 + 1$ unconstrained variables $\lambda(i)$, $\mu(j)$, and x. In the following we will show how finding the extremum of this function allows us to solve the augmented assignment problem.

C. Optimizing the effective free energy

Let $\overline{G}(i, j)$, $\overline{n_1}(i)$, and $\overline{n_2}(j)$ be the expected values of $G(i, j)$, $n_1(i)$, and $n_2(j)$, respectively, with respect to the Gibbs distribution given in Eq. (4). We use a saddle point approximation (SPA) to compute those values, namely we compute extrema of the effective free energy with respect to the variables λ μ, and x:

$$\begin{aligned} &\frac{\partial F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)}{\partial \lambda_i} = 0, \quad \frac{\partial F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)}{\partial \mu_j} = 0, \\ &\frac{\partial F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)}{\partial x} = 0. \end{aligned} \quad (9)$$

After some rearrangements, those two equations can be written as

$$\forall i, \sum_j X(i, j) = d_1(i), \quad (10a)$$

$$\forall j, \sum_i X(i, j) = d_2(j), \quad (10b)$$

$$\sum_{i,j} X(i, j) = k, \quad (10c)$$

where

$$\begin{aligned} X(i, j) &= h[\beta(C(k, l) + i\lambda(i) + i\mu(j) + ix)], \\ d_1(i) &= h(-i\beta\lambda(i)), \\ d_2(j) &= h(-i\beta\mu(j)), \end{aligned} \quad (11)$$

and

$$h(x) = \frac{1}{e^x + 1}. \quad (12)$$

Note that this system is based on $N_1 + N_2 + 1$ variables, the $\lambda(i)$ for $i \in [1, N_1]$, $\mu(j)$ for $j \in [1, N_2]$, and x. As is often the case, the saddle-point may be purely imaginary. In the present case, one can easily see from Eq. (10) that the variables $i\lambda(i)$, $i\mu(j)$, and ix must be real and in the following, we will replace $\{i\lambda(i), i\mu(j), ix\}$ by $\{\lambda(i), \mu(j), x\}$.

To analyze the SPA, we need to check the existence and assess the unicity of the critical points of the free energy. The following theorem shows that $F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)$ is strictly concave.

Theorem 1. The Hessian of the effective free energy $F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)$ is negative definite. Therefore, the free-energy function is strictly concave.

Proof. See Appendix B. ■

We have the following property that relates the solutions of the SPA system of equations to the expected values for the transportation plan and indicator functions:

Property 1. Let $\overline{S}_\beta$ be the expected state of the system at the temperature β with respect to the Gibbs distribution given in Eq. (4). $\overline{S}_\beta$ is associated with an expected transportation plan $\overline{G}_\beta$ and expected indicator functions $\overline{n_1}_\beta$ and $\overline{n_2}_\beta$. Let $\lambda^{MF}(i)$, $\mu^{MF}(j)$, and x^{MF} be the solutions of the system of Eqs. (10). Then the following identities hold:

$$\begin{aligned} \overline{G}_\beta(i, j) &= h(\beta(C(i, j) + \lambda^{MF}(i) + \mu^{MF}(j) + x^{MF})), \\ &= X^{MF}(i, j), \\ \overline{n}_{1\beta}(i) &= h(-\beta\lambda^{MF}(i)) = d_1^{MF}(i), \\ \overline{n}_{2\beta}(j) &= h(-\beta\mu^{MF}(j)) = d_2^{MF}(j). \end{aligned} \quad (13)$$

Note that the solutions are mean-field solutions, hence the superscript MF.

Proof. See Appendix C. ■

For a given value of the parameter β, $\overline{n}_{1\beta}$ and $\overline{n}_{2\beta}$ are indicators of the elements of S_1 and S_2 that are in correspondence and $\overline{G}_\beta$ forms a transportation plan between S_1 and S_2 that is optimal with respect to the free energy defined in Eq. (8). Note that these values are mean values and possibly fractional. They will only be exactly 0 or 1 at $\beta = +\infty$, i.e., at 0 temperature. We can associate to this transportation plan an optimal free energy F_β^{MF} and an optimal internal energy $U_\beta^{\mathrm{MF}} = \sum_{i,j} \overline{G}_\beta(i, j)C(i, j)$. Those two values are the mean-field approximations of the exact free energy and internal energy of the system, respectively. We now list important properties of U_β^{MF} and F_β^{MF}:

Property 2. F_β^{MF} and U_β^{MF} are, respectively, monotonic increasing and monotonic decreasing functions of the parameter β.

Proof. See Appendix D for F_β^{MF} and Appendix E for U_β^{MF}. ■

Theorem 1 and Property 2 highlight a number of advantages of the proposed framework that rephrases the unbalanced assignment problem as a temperature dependent process. First, at each temperature the unbalanced k-cardinality assignment problem is turned into a strongly concave problem with a unique solution. This problem has a linear complexity in the number of variables, compared to the quadratic complexity of the original problem. The concavity allows for the use of simple algorithms for finding a minimum of the effective free-energy function [Eq. (8)]. We note also that Eqs. (13) provides good numerical stability for computing the transportation plan and the indicator functions n_1 and n_2, because of the behavior of the function $h(x)$ (see below). Finally, the convergence as a function of temperature is monotonic.

D. Rewriting the free energy

Equation (8) provides an expression for the free energy of the system as a function of the unconstrained variables $\lambda(i)$, $\mu(j)$, and x. As written, this free energy does not have a standard form as seen in thermodynamics as it does not include the corresponding energy U, nor does it define an entropy S (both are functions of the same unconstrained variables. We derive a new form for the free energy. The internal energy is

$$U_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x) = \sum_{i,j} C(i,j)X(i,j),$$

where $X(i,j)$ is defined in Eq. (11). We define the function f for $x \in (0,1)$:

$$f(x) = -x\ln(x) - (1-x)\ln(1-x). \tag{14}$$

We have the following property:

Theorem 2. The effective free energy of the assignment problem can be written as

$$\begin{aligned} &F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x) \\ &\quad = U_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x) - TS_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x) \\ &\qquad + x\left(\sum_{i,j} X(i,j) - k\right) + \sum_i \lambda(i)\left(\sum_j X(i,j) - d_1(i)\right) \\ &\qquad + \sum_j \mu(j)\left(\sum_i X(i,j) - d_2(j)\right), \end{aligned} \tag{15}$$

where we have defined the entropy S as

$$\begin{aligned} &S_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x) \\ &\quad = \sum_{i,j} f(X(i,j)) + \sum_i f(d_1(i)) + \sum_j f(d_2(j)), \end{aligned} \tag{16}$$

where X, d_1, and d_2 are defined in Eq. (11)/ In particular, at the maximum of the free energy,

$$\begin{aligned} U_\beta^{MF} &= \sum_{i,j} C(i,j)\overline{G}_\beta(i,j), \\ S_\beta^{MF} &= \sum_{i,j} f(\overline{G}_\beta(i,j)) + \sum_i f(\overline{n}_{1\beta}(i)) + \sum_j f(\overline{n}_{2\beta}(j)), \\ F_\beta^{MF} &= U_\beta^{MF} - TS_\beta^{MF}. \end{aligned} \tag{17}$$

Proof. See Appendix F. ∎

The form of the free energy given in Eq. (15) has an intuitive physical interpretation. The first term is the original k-cardinality assignment energy, the second is $-T$ times an entropy term, and the third, fourth, and fifth terms impose the constraints via Lagrange multipliers. At the saddle point, those constraints are satisfied, and the free energy has its generic form of energy minus temperature times entropy. The entropy term involves the function f applied on the converged $\overline{G}_\beta(i,j)$, $\overline{n}_{1\beta}(i)$, and $\overline{n}_{2\beta}(j)$. This function can be seen as a double entropic barrier, imposing that those variables remain within (0,1). It is only at $T = 0$ ($\beta = +\infty$) that they will take either the value 0 or 1.

III. SOLVING THE GENERIC k-ASSIGNMENT PROBLEM

In the previous section, we have described a formalism based on statistical physics for solving the unbalanced k-cardinality assignment problem. We have derived an effective free energy, $F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)$, that depends on $N_1 + N_2 + 1$ unconstrained variables $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$, and x. We have shown that this free energy is strictly concave and that its maximum is found by solving a system of nonlinear equations, at each inverse temperature β. We have also shown that the trajectory of the maxima $F^{\mathrm{MF}}(\beta)$ as a function of β is monotonic, increasing. We need to establish now that this trajectory allows us to find the actual solution of the unbalanced k-cardinality assignment problem. Recall that this solution is defined by a transportation matrix G^* and its corresponding energy U^*. In this section, we will assume that the assignment problem is nondegenerate and that it has a unique solution. We will fully characterize what it means in the next section.

We first prove that the optimal assignment energy U^*, is equal to the infinite inverse temperature limit of both the mean-field free energy and the internal energy:

Theorem 3.

$$\begin{aligned} U^* &= \lim_{\beta\to+\infty} F_\beta^{\mathrm{MF}}, \\ U^* &= \lim_{\beta\to+\infty} U_\beta^{\mathrm{MF}}. \end{aligned} \tag{18}$$

Proof. See Appendix G. ∎

As the trajectories of F_β^{MF} and U_β^{MF} as a function of β were already found to be respectively monotonically increasing and monotonically decreasing, this theorem adds the information that at the infinite inverse temperature limit (or equivalently at the zero-temperature limit), both converge to the optimal assignment energy. These results validate our statistical physics approach and the saddle-point approximation. Note however that they define the behavior of the energy and free energy, and not of the coupling matrix $\bar{G}_\beta = X^{\mathrm{MF}}$. As $\overline{G}_\beta(i,j) = h(\beta(C(i,j) + \lambda^{\mathrm{MF}}(i) + \mu^{\mathrm{MF}}(j) + x^{\mathrm{MF}}))$ and $0 < h(x) < 1$, the coupling matrix at a finite temperature is fractional. We need to show that as $\beta \to +\infty$, the corresponding matrix $\overline{G}_\infty$ does converge to the partial permutation matrix G^*, and not to a fractional matrix that would lead to the same low energy U^*.

We first establish bounds on the entropy, internal energy, and free energy at the SPA. Let us define $A(N_1, N_2) = N_1N_2\ln(2) + N_1\ln(2) + N_2\ln(2)$; then

Theorem 4.

$$0 \leqslant S_\beta^{\mathrm{MF}} \leqslant A(N_1, N_2), \tag{19}$$

$$U^* - \frac{A(N_1,N_2)}{\beta} \leqslant F_\beta^{\mathrm{MF}} \leqslant U^*, \tag{20}$$

$$U^* \leqslant U_\beta^{\mathrm{MF}} \leqslant U^* + \frac{A(N_1,N_2)}{\beta}. \tag{21}$$

Proof. See Appendix H. ∎

The two previous theorems are valid for all unbalanced k-cardinality assignment problems. We establish now bounds on the element of the assignment matrix $\overline{G}_\beta$ *in the specific case that this k-cardinality assignment problem has a unique solution*. The matrix $\overline{G}_\beta$ denotes the unique doubly substochastic matrix associated with the minimum of the free energy at

the inverse temperature β. The next theorem bounds how close this doubly substochastic matrix is to the *unique* partial permutation matrix, G^*, representing the optimal solution to the unbalanced assignment problem.

Theorem 5. Suppose that the unbalanced k-cardinality assignment problem associated with the $N_1 \times N_2$ cost matrix C admits a unique optimal partial assignment matrix, G^*. Let Δ be the difference in total cost between the optimal solution and the second-best solution. Then,

$$\max_{i,j} |\overline{G}_\beta(i,j) - G^*(i,j)| \leqslant \frac{A(N_1,N_2)}{\beta\Delta}. \tag{22}$$

Proof. See Appendix I. ■

This theorem validates that in the generic case in which the solution to the unbalanced k-cardinality assignment problem is unique, the converged solution matrix $\overline{G}_\infty$ is this unique solution to the unbalanced k-cardinality assignment problem, G^*. In addition, it provides bounds to how close $\overline{G}_\beta$ is from the optimal solution at any inverse temperature β. For example,

Theorem 6. Suppose that the unbalanced k-cardinality assignment problem associated with the $N_1 \times N_2$ cost matrix C admits a unique optimal partial assignment matrix, G^*. Let Δ be the difference in total cost between the optimal solution and the second-best solution. Then, rounding-off each of the entries of $\overline{G}_\beta$ to the nearest integer yields the partial permutation matrix G^* that solves the k-assignment problem whenever

$$\beta > \frac{2A(N_1,N_2)}{\Delta}.$$

The proof follows directly from Theorem 5 and from the fact that rounding off to the nearest integer will yield the optimal assignment matrix whenever,

$$\max_{i,j} |\overline{G}_\beta(i,j) - G^*(i,j)| < \tfrac{1}{2}.$$

We conclude that in the generic case we can solve the unbalanced k-cardinality assignment problem exactly at finite, although sufficiently high inverse temperature β. We should highlight, however, that Theorem 6 is not easy to implement as it is difficult to estimate Δ. As an alternate to this theorem, we propose the following theorem:

Theorem 7. Suppose that the unbalanced k-cardinality assignment problem associated with the $N_1 \times N_2$ cost matrix C admits a unique optimal partial assignment matrix, G^*. Let us assume that at an inverse temperature β, the current solution matrix $\overline{G}_\beta$ contains exactly k values that are greater or equal to $\frac{1}{2}$. Then, rounding-off each of the entries of $\overline{G}_\beta$ to the nearest integer yields the partial permutation matrix G^* that solves the unbalanced k-cardinality assignment problem.

Proof. See Appendix J. ■

This theorem defines a criterion that is easily implemented (we will see below) to terminate the annealing process in β when solving the unbalanced k-cardinality assignment problem with our method.

IV. IMPLEMENTATION

We have implemented the unbalanced k-cardinality assignment framework described here in a C++ program UMatching that is succinctly described in Algorithm 1.

ALGORITHM I. UMatching: a temperature dependent framework for solving the unbalanced k-cardinality assignment problem.

Input: The sizes associated to the problem: N_1 and N_2, the number of agents and tasks, and k, the expected number of assignments; the cost matrix C. Initial value β_0 for β
Initialize: Initialize arrays $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ to 0 and initialize $x = 0$. Set $STEP = \sqrt{10}$.
for $i = 1, \ldots$ until convergence **do**
(1) Initialize $\beta^i = STEP * \beta^{i-1}$.
(2) Solve nonlinear Eqs. (10) for $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$, and x at saddle point
(3) Compute corresponding $\overline{G}_\beta$, $\overline{n}_{1\beta}$, $\overline{n}_{2\beta}$, and U_β^{MF}
(4) Check for convergence: if $\overline{G}_\beta$ contains exactly k values that are greater than 0.5, stop
end for
Output: The converged assignment matrix $\lfloor \overline{G}_\beta \rceil$, the indicator functions $\lfloor \overline{n}_{1\beta} \rceil$, $\lfloor \overline{n}_{2\beta} \rceil$ over the agents and tasks, and the minimal associated cost U_β.

UMatching is based on an iterative procedure in which the parameter β (inverse of the temperature) is gradually increased. At each value of β, the nonlinear system of equations defined by Eq. (10) is solved. We write this system as

$$\mathbf{A}_\lambda = 0, \quad \mathbf{A}_\mu = 0, \quad A_x = 0,$$

where $\mathbf{A} = (\mathbf{A}_\lambda, \mathbf{A}_\mu, A_x)$ is a vector of predicates defined as

$$A_\lambda(i) = \sum_j \frac{1}{e^{\beta(C(i,j)+\lambda(i)+\mu(j)+x)}+1} - \frac{1}{e^{-\beta\lambda(i)}+1},$$

$$A_\mu(j) = \sum_i \frac{1}{e^{\beta(C(i,j)+\lambda(i)+\mu(j)+x)}+1} - \frac{1}{e^{-\beta\mu(j)}+1},$$

$$A_x = \sum_{i,j} \frac{1}{e^{\beta(C(i,j)+\lambda(i)+\mu(j)+x)}+1} - k.$$

This system has $N_1 + N_2 + 1$ equations, with the same number of variables. It is solved using an iterative Newton-Raphson method (for details, see, for example, Refs. [8,15]). Once the SPA system of equations is solved, the assignment matrix $\overline{G}_\beta$, the indicator functions $\overline{n}_{1\beta}$, $\overline{n}_{2\beta}$ and the corresponding transportation energy $U^{\mathrm{MF}}(\beta)$ are computed. Using Theorem 7, the iterations over β are stopped if the matrix $\overline{G}_\beta$ contains exactly k values that are larger than 0.5. Otherwise, β is increased, and the current values of $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$ and x are used as input for the following iteration. At convergence, the values of the assignment matrix and indicator functions are rounded to the nearest integer (indicated as $\lfloor \rceil$ in the output of Algorithm 1). The minimal energy is then computed using the corresponding integer matrix.

As for any annealing scheme, the initial temperature, or, in our case, the initial value β_0 is a parameter that significantly impacts the efficiency of the algorithm. Setting β_0 to be too small (i.e., a high initial temperature) will lead to inefficiency as the system will spend a significant amount of time at high temperatures, while setting β_0 too high will require many steps to converge at the corresponding low temperature, thereby decreasing the efficiency brought by annealing. The value of β scales the cost matrix C and as such is related to the range of this matrix, more specifically to its largest value,

$C_{\max}$. We found that setting $\beta_0 C_{\max} = 1$ provides satisfactory annealing efficiency for all test cases presented in the numerical simulation sections. We cannot exclude that there are cases for which this settng is not optimal. This is a general concern with annealing procedures.

The main computing cost of this algorithm is associated with solving the nonlinear set of equations corresponding to the SPA at each value of β. As the free-energy function associated with this system is concave (see Theorem 1), we use the Newton-Raphson method. This method works by iteratively linearizing the system of equations. It therefore requires solving linear systems of equations. We have implemented the MINRES [17] iterative method to solve those linear system. This method, as well as all the linear algebra involved in the algorithm can easily be implemented on a graphics processor unit (GPU), using the CUDA toolkit and the associated optimized cuBLAS library [18].

V. SOLVING DEGENERATE ASSIGNMENT PROBLEMS

Our statistical physics approach is basically a relaxation approach to the unbalanced assignment problem. Indeed, we build a collection of real matrices $\bar{G}_\beta$ that minimizes the assignment cost and that are doubly substochastic. Those matrices are strictly doubly substochastic, i.e., their entries and non integer values in the interval (0,1). If the unbalanced k-cardinality assignment problem is known to have a unique integer solution, then we have shown that those matrices converge to a binary partial permutation matrix G^* that solves the problem when $\beta \to +\infty$. We have even established a criterion for terminating the numerical procedure to reach that convergence. The question remains as to what happens when the problem is degenerate, i.e., when it may have multiple integer solutions.

The unbalanced k-cardinality assignment problem is a linear programming problem. Checking if such a problem is degenerate is unfortunately often NP complete [19,20]. The degeneracies occur due to the presence of cycles in the linear constraints, i.e., in the cost matrix for the assignment problem. If this is the case, then we propose to randomly perturb that matrix to bring it back to the generic problem. Megiddo and Chandrasekaran [21] have shown that a ε-perturbation of a degenerate linear programming problem reduces this problem to a nondegenerate one. We state this result as follows for the unbalanced assignment problem:

Property 3. Suppose that the solution $\overline{G}_\beta$ to the unbalanced k-cardinality assignment problem associated with the $N_1 \times N_2$ cost matrix C has a nonzero entropy $S^{MF}(\beta)$ when $\beta \to +\infty$. Let Δ be the difference in total cost between the optimal solution and the second-best solution. Then, adding random uniform noise with support $[0, \alpha]$ to each value of C and solving the unbalanced assignment problem on this perturbed matrix will generate one integer solution that is also a solution to the unperturbed unbalanced assignment problem, whenever $\alpha < \frac{\Delta}{2k}$.

Proof. The result on the use of a perturbation to get a nondegenerate solution follows from Ref. [21]. The result on the bound for α is very similar to the equivalent result for the balanced assignment problem that can be found in Appendix H of Ref. [16]. ■

This proposition gives us a general strategy for solving a minimum cost unbalanced k-cardinality assignment problem for any cost matrix C:

(a) Solve the unbalanced k-cardinality assignment problem using the statistical physics approach described in the previous section. If the entropy converges to 0 as $\beta \to +\infty$, then the solution is guaranteed to be a partial permutation. This matrix can be derived by rounding-off each element of $\overline{G}_\beta$ whenever it contains exactly k values that are greater than 0.5.

(b) If the approach described in option (a) fails, i.e., the entropy does not converge to 0 or $\overline{G}_\beta$ never contains k values that are greater than 0.5, then the problem is deemed degenerate. To remove this degeneracy, we propose to scale the values of the cost matrix so that they become integer, in which case Δ is necessarily an integer value. We then start by assuming that $\Delta = 1$ and set $\alpha = 1/(2k)$. We then repeat the optimization of the transportation plan. If it still does not converge [i.e., based on the criteria of option (a)], then we can increase α by a factor of 2 iteratively, until the system converges, The solution to this perturbed problem will also be a solution to the original problem.

VI. NUMERICAL SIMULATIONS

In this section we describe numerical experiments designed to illustrate the behavior and assess the validity and convergence of our algorithm (A), as well as its efficiency compared to other generic algorithms for solving balanced and k-cardinality assignment problems (B).

A. Correctness of our algorithm

The next three subsections relate to assessment for continuous random cost matrices, for discrete random cost matrices, and for special matrices corresponding to hard assignment problems for the Hungarian algorithm [22,23], respectively.

1. Random k-cardinality assignment problems with continuous cost matrices

Results of assignment problems are anecdotic in the sense that they are problem dependent. Random assignment problems are exceptions, however, as they are many theoretical results associated with them (for review, see Ref. [24]). Let us consider the k-cardinality assignment problem between two sets of points of size N_1 and N_2, respectively. If the elements $C(i, j)$ of the cost matrix are independent and identically distributed (iid) exponential with mean of 1, then the expected value of the minimum k-assignment cost, $U^*_{N_1,N_2}$ has the form

$$E\left[U^*_{N_1,N_2}\right] = \sum_{\substack{i,j\geqslant 0 \\ i+j<k}} \frac{1}{(N_1 - i)(N_2 - j)}, \tag{23}$$

where $(i, j) \in [1, N_1] \times [1, N_2]$. This equality was first conjectured and verified for the cases $k = 1, k = 2, k = N_1 = 3, k = N_1 = N_2 = 4$ by Ref. [25]. It was validated again by Ref. [26] for $k \leqslant 4, k = N_1 = 5$, and $k = N_1 = N_2 = 6$, and finally proven independently by Refs. [27,28] (see also Ref. [29]). Note that in the special case $k = N_1 = N_2$, the

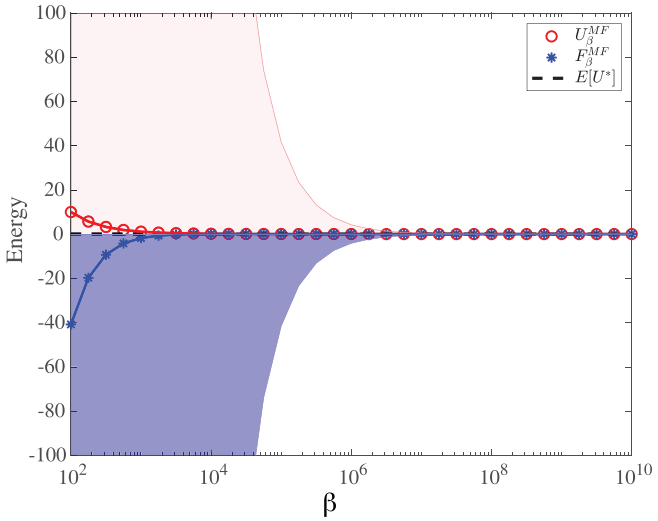

FIG. 1. Convergence of the internal energy U_β^{MF} (circle, red) and free energy F_β^{MF} (star, blue) as a function of β when solving a random k-assignment problem with a cost matrix C of size 2000×3000 whose elements are independent identically distributed values drawn from exponential distributions with mean 1, with $k = 1000$. We show the bounds on the internal energy and on the free energy computed from the theoretical bounds given in Theorem 4 as a light shaded red patch (positive part of the plot) and as a shaded blue patch (negative part of the plot), respectively. The dotted horizontal line show the expected value for the minimal cost of an exponential random k-assignment problem of the same size.

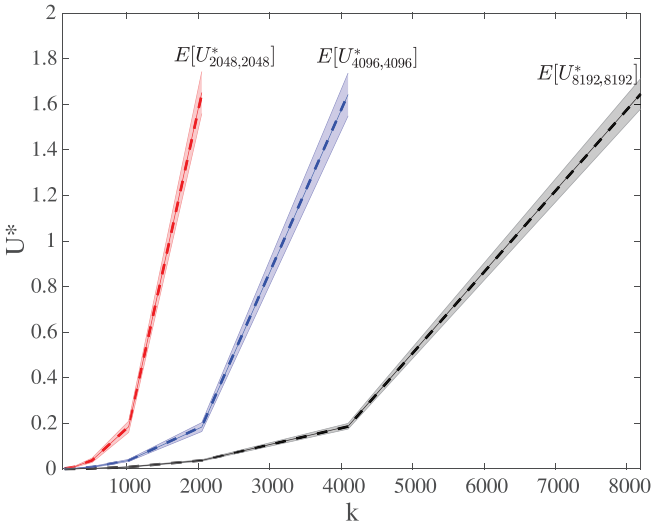

FIG. 2. The converged internal energy $U_{+\infty}^{\mathrm{MF}}$ as a function of k when solving a random k-cardinality assignment problem with a cost matrix C of size 2048×2048 (left, red), 4096×4096 (center, blue), and 8192×8192 (right, black), whose elements are independent identically distributed values drawn from exponential distributions with mean 1. At each value of k, 10 experiments were performed; the mean values are shown as a solid line and the corresponding standard deviations are illustrated with a shaded area. The corresponding theoretical values given by Eq. (23) are shown as dashed lines; in all three cases, they overlap with the mean value lines.

equality is equivalent to

$$E\left[U^*_{N_1,N_2}\right] = \sum_{i=1}^{N_1} \frac{1}{i^2}.$$

This equality for the balanced assignment problem was initially conjectured by Parisi [30] and proved by Linusson and Wästlund [27]. We note that such random problems are guaranteed to have a unique solution: as the elements of the cost matrix are iid variables, there is a zero probability that they can form cycles; the corresponding assignment problem has a unique solution matrix whose entries are 0 or 1.

As a first illustration of our procedure, we ran UMatching on a random cost matrix with exponential distributions with mean 1 of size 2000×3000, solving the k-assignment problem with $k = 1000$. In Fig. 1, we show the corresponding trajectories of the internal energy U_β^{MF} and free energy F_β^{MF} as well as the theoretical bounds on those values given in 4. As expected, the internal energy is monotonically decreasing while the free energy is monotonically increasing, and both converge to the same value, 0.119. Note that from Eq. (23), the expected value of the minimum cost associated with a matrix of this size and $k = 1000$ is $E[U^*] = 0.115$, i.e., very close to the value observed with the specific cost matrix that was generated for this example. Note also that both the internal energy and the free energy have basically converged for $\beta > 10^7$. This is confirmed as we found that the matrix $\overline{G}_\beta$ contained exactly $k = 1000$ values that are greater than 0.5 when $\beta > 10^7$, i.e., the procedure could have been stopped then.

Our second experiment considers three types of random cost matrices with exponential distributions with mean 1 of size $N \times N$, with $N = 2048, 4096$, and 8192, respectively. For each matrix we solve the k-cardinality assignment problem for values of k varying from 128 to N. For each value of k, we ran 10 different random instances and computed the mean value and standard deviation of the corresponding converged minimal costs, U^*, where convergence is detected based on Theorem 7. In Fig. 2, we plot the converged energies as a function of k for each value of N, as well as the corresponding theoretical values for the expectancy of U^*, given by Eq. (23). Note that the curves of the mean values of the converged energy and of the theoretical values for the expectancies of those energies overlap, showing full agreement.

2. Random balanced assignment problems with discrete cost matrices

One of the advantages of using continuous random cost matrices is that the assignment problem is nondegenerate. Discrete random cost matrices offer an interesting departure from this, as they are likely to include cycles and therefore to lead to degenerate assignment problems. To assess the ε-perturbation approach described in the previous section is able to solve such degenerate problems, we considered the four types of discretely distributed random cost matrices C of size $N \times N$ considered by Parviainen [31]:

(a) Each row of C^a is an independent random permutation of the set $\{1, 2, \ldots, N\}$, chosen uniformly from the set of permutations.

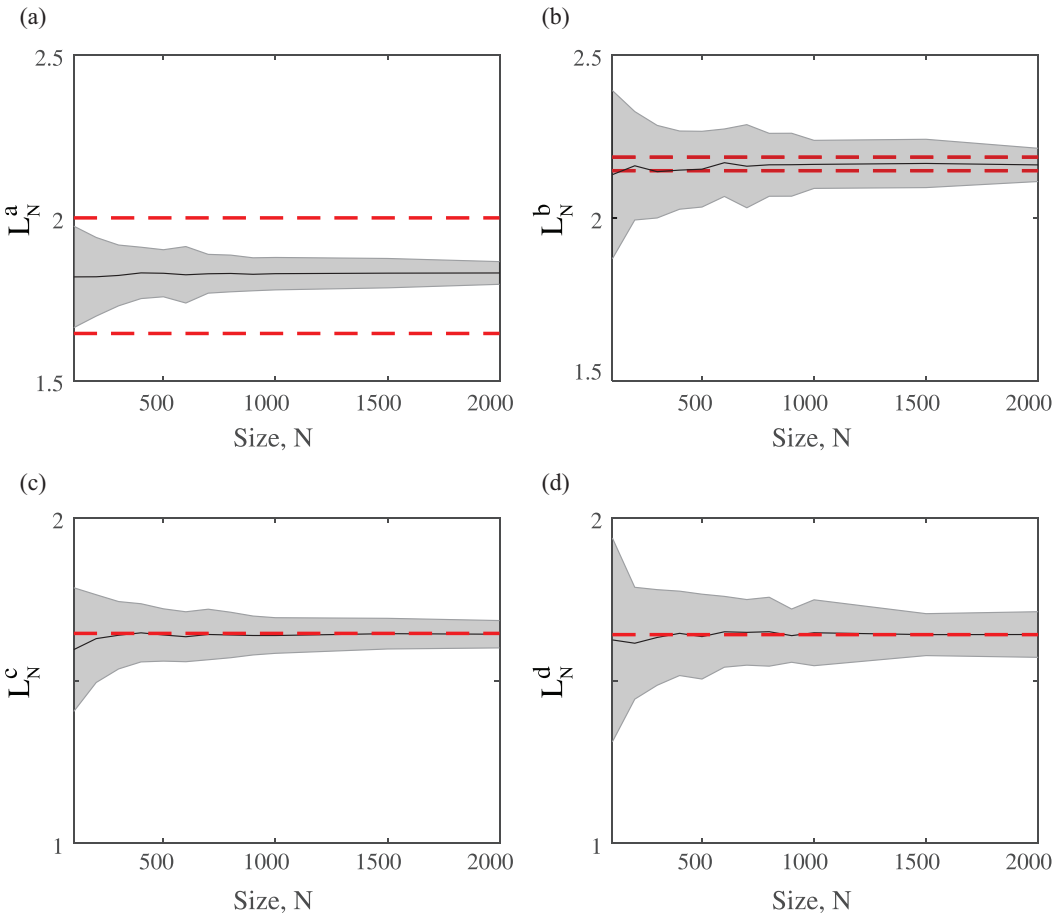

FIG. 3. The converged normalized minimal cost as a function of N when solving a discrete random assignment problem with a cost matrix of size $N \times N$ whose elements are randomly selected based on schemes (a) [panel (a)], (b) [panel (b)], (c) [panel (c)], and (d) [panel (d)] (see text for details on those schemes). At each value of N, 100 experiments were performed; the mean values are shown as a solid line and the corresponding standard deviations are illustrated with a grey shaded area. The limits on corresponding theoretical values given by Eq. (24) are shown as dashed red lines.

(b) Each element of C^b is an independent random number chose uniformly in $\{1, 2, \ldots, N\}$.

(c) C^c is an independent random permutation of the set $\{1, 2, \ldots, N^2\}$, chosen uniformly from the set of permutations.

(d) Each element of C^d is an independent random number chose uniformly in $\{1, 2, \ldots, N^2\}$.

Note that all these problems are balanced, with $k = N$.

In all four problems, the minimal cost is a function of N. To remove this dependence, we normalize the cost as follows:

$$L_N^a = \frac{1}{N} U(G^*, C^a),$$
$$L_N^b = \frac{1}{N} U(G^*, C^b),$$
$$L_N^c = \frac{1}{N^2} U(G^*, C^c),$$
$$L_N^d = \frac{1}{N^2} U(G^*, C^d),$$

where the superscripts (a, b, c, and d refer to the type of discrete assignment problem considered. Parviainen [31] established the following properties of the expected values for the minimal costs when $N \to +\infty$ (i.e., $E[L] = \lim_{N\to+\infty} E[L_N]$):

$$\frac{\pi^2}{6} \leqslant E[L^a] \leqslant 2,$$
$$\frac{\pi^2}{6} + \frac{12}{24} \leqslant E[L^b] \leqslant \frac{\pi^2}{6} + \frac{13}{24}, \tag{24}$$
$$E[L^c] = E[L^d] = \frac{\pi^2}{6}.$$

We ran four type computational experiments, one for each type of discrete random cost described above. In each experiment, we considered 100 random discrete cost matrices. For each matrix, we solved the assignment problem using UMatching for values of N varying from 100 to 2000 using an ε-perturbation of the cost matrix, with $\varepsilon = 5 \times 10^{-4}$. We computed the mean value and standard deviation of the corresponding converged normalized minimal costs, L_N, where convergence is detected based on Theorem 7. In Fig. 3, we plot the normalized costs, L_N^a, L_N^b, L_N^c, L_N^d as a function of

N. We also plotted the boundaries for the corresponding limit expected values, as given by Eqs. (24). Note that the agreement between the computed mean and the theoretical limits improve as N increases.

3. Hard assignment problem: Machol-Wien cost matrices

Let us consider the special assignment problem between two sets of points of cardinality N, with a cost matrix referred to as the Machol-Wien cost matrix [23,32] (note that this matrix was originally proposed by Silver [22]):

$$C(i,j) = (i-1) \times (j-1) \quad \forall (i,j) \in [1,N]^2.$$

The unique optimal solution to the corresponding balanced assignment problem is $G(i,j) = 1$ if $i + j = n + 1$ and $G(i,j) = 0$ otherwise, with $(i,j) \in [1,N]^2$ with a total cost W given by

$$W_N = \frac{N(N-1)(N-2)}{6}. \tag{25}$$

This type of cost matrices was designed to be hard for the Hungarian algorithm as it leads to a worst-case scenario [22,23]. In a survey of multiple algorithms and codes for solving dense assignment problems, Dell'Amico and Toth [33] showed that assignment problems based on these matrices are very difficult for all the methods they have tested.

Equation (25) can be extended to the case of the k-cardinality assignment problem,

$$W_{N,k} = \frac{k(k-1)(k-2)}{6}. \tag{26}$$

Compared to the balanced Machol-Wien problem, the corresponding k cardinality problem is degenerate and does not have a unique solution. There are in fact $N - k + 1$ degenerate solutions with the same energy given by Eq. (26). The Mth solution ($1 \leqslant M \leqslant N - k + 1$) is given by

$$G(i,j) = \begin{cases} 1 & \text{if } i + j = k + 1, \\ 1 & \text{if } i = M + k,\ j = 1, \\ 0 & \text{otherwise,} \end{cases}$$

with $(i,j) \in [1,N]^2$. Note that Eq. (26) is independent of N.

To verify numerically Eq. (26), we considered Machol-Wien cost matrices of size $N \times N$, with $N = 8192$. For each matrix we solve the k-cardinality assignment problem for values of k varying from 128 to N. In Fig. 4, we plot the converged energies as a function of k (red circles), as well as the corresponding theoretical values for the expectancy of U^* (solid black line), given by Eq. (26). Note the full agreement.

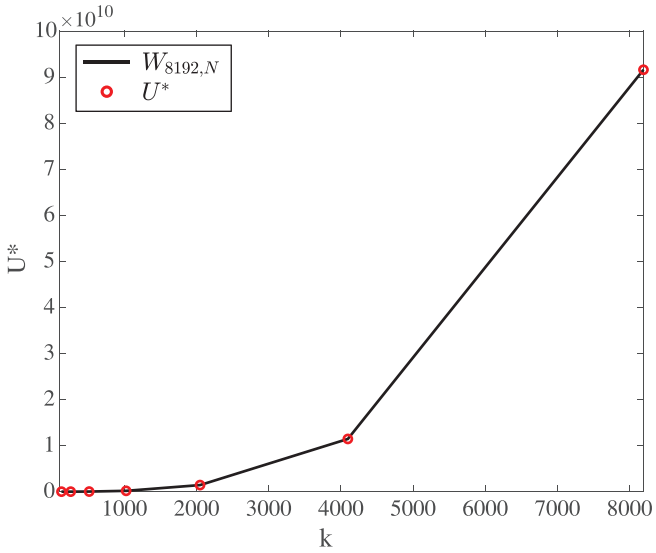

FIG. 4. The converged internal energy $U^*_{+\infty}$ as a function of k when solving a k-cardinality assignment problem for the Machol-Wien cost matrix C of size 8192×8192. At each value of k, the computed value is shown as a red circle and the corresponding theoretical value [Eq. (26)] shown as a solid black line.

B. Computing efficiency of UMatching

We have developed a method for solving the k-cardinality assignment problem that extends to the balanced and unbalanced assignment problems. We have claimed that this method provides fast and robust solutions to those assignment problems. To check that it is indeed the case, we have benchmarked its running times with those of existing solutions for solving balanced assignment problems as well as k-cardinality assignment problems. We have implemented two versions of UMatching, one that runs on CPUs, and another that runs on GPUs.

For the CPU version of UMatching, we compared it with 2 different codes that solve the balanced assignment problem using the Hungarian algorithm, which we refer to as LAP and LAPJV, respectively, and with one program that solves the k-cardinality assignment problem, SKAP.

The Hungarian algorithm remains a standard for solving balanced as well as unbalanced assignment problems. This algorithm is either order $O(N^4)$ or $O(N^3)$ for the balanced problem of size N, depending on its implementation. A typical $O(N^4)$ implementation is based on a matrix formulation of the assignment problem: it follows the original idea of Munkres [34] and is described in detail in a tutorial by Pilgrim [35]. We used a Fortran-90 implementations of those ideas, available at [36]. Better (in terms of computing complexity) implementations of the Hungarian algorithm follow its graph formulation. Such implementations are global and are based on improving a matching along augmenting paths (i.e., alternating paths between unmatched vertices). They are order $O(N^3)$. We used the version originally proposed by Jonker and Volgenant [37] and available at [38]. Finally, Dell'Amico, Lodi, and Martello developed specialized code to solve the k-cardinality assignment problems for dense [2] as well as for sparse [39] graphs. Their code, SKAP, handles both types of graph and is able to solve balanced and unbalanced assignment problems, as well as k-cardinality assignment problems. It is available at [40]. Note that this code is specific to integer cost matrices.

For the GPU version of UMatching, we considered two different GPU implementations of the Hungarian algorithm. There are many efforts aimed at parallelizing this algorithm (see, for example, Refs. [6,7,41] and references therein). The first implementation we considered, HunCUDA, is based on

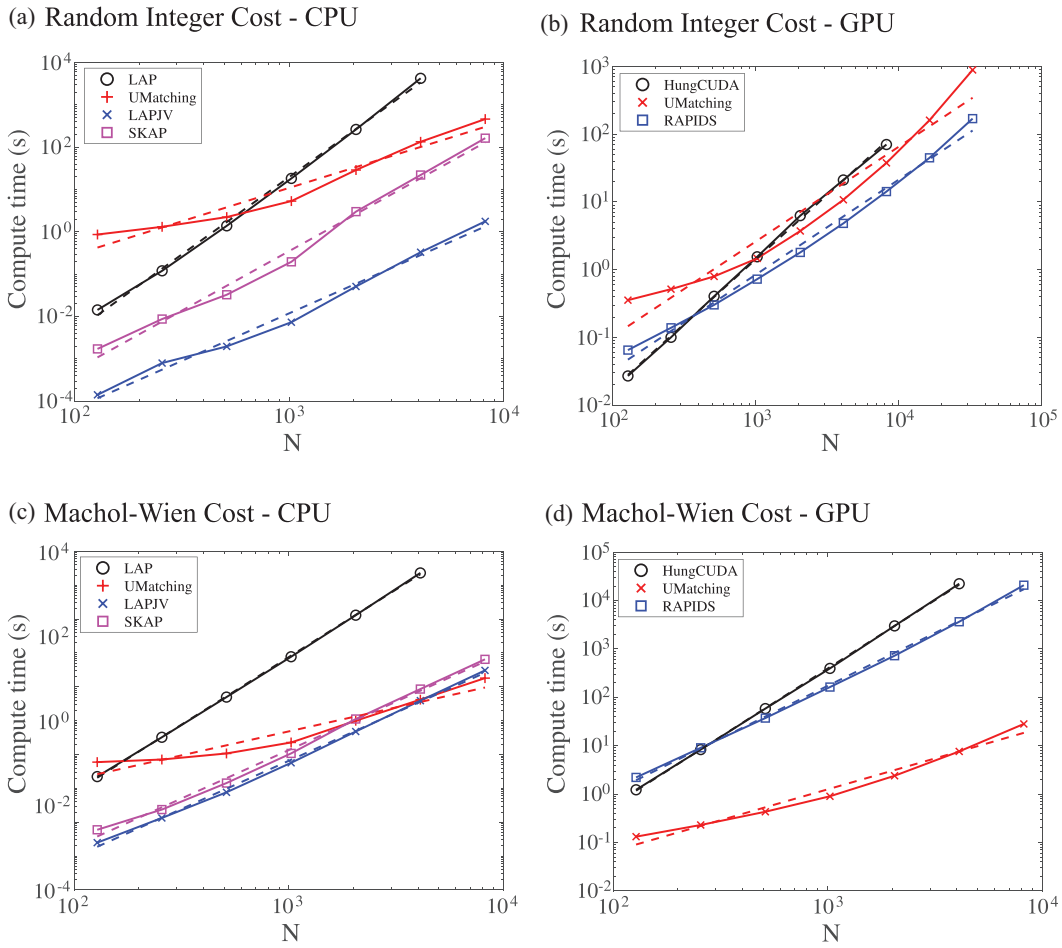

FIG. 5. Solving integer assignment problems. The computing times based on CPU (a), (c) and on GPU (b), (d) implementations of different programs designed for solving balanced assignment problems. We consider integer random balanced problems of size N (a), (b) as well as integer Machol-Wien problems of size N (c), (d) (see text for details). On CPU, four programs are tested: LAP (circle, black) and LAPJ (x, blue), two different implementations of the Hungarian algorithm, SKAP (square, magenta), a specialized program for the k-cardinality assignment problem, and UMatching (+, red), introduced in this study. On GPU, three programs are tested: HungCUDA (circle, black), RAPIDS (square, blue), and UMatching (x, red). See text for a brief introduction on all these methods. The mean values over 10 different runs are plotted as solid lines, while the dotted lines represent linear fits in the log domain. The slopes of those lines, which correspond to the observed computational complexity, are given in Table I. All computations were run on a computer with an AMD Ryzen Threadripper PRO 3975WX with 32 CPUs (64 cores) and a NVIDIA GPU RTX A5000.

the parallel version developed using CUDA by Lopes *et al.* [41] and available at [42]. The key feature of HunCUDA is an implementation of the alternating path search phase of the Hungarian algorithm that is distributed by several blocks, thereby minimizing global device synchronization. Note that this implementation is specific to balanced assignment problems whose sizes are powers of two. The second implementation considered, which we refer to as RAPIDS, is based on the implementation of the Hungarian algorithm in the recent package cugraph that is itself part of the suite of program RAPIDS [43] developed by NVIDIA for data science pipelines on the GPU. It is available in open-source format at [44]. We used the C++ implementation of cugraph.

All benchmarks were run on a computer with an AMD Ryzen Threadripper PRO 3975WX with 32 CPUs (64 cores) and a NVIDIA RTX A5000. Note that the computing times vary little with k, with larger values for k, respectively, small and large.

1. Comparing UMatching with fast implementations of the Hungarian algorithms for balanced assignment problems

We first tested the different programs described above (both CPU and GPU based) on balanced assignment problems based on random matrices. As SKAP only applies to integer cost matrices, our test sets includes the C^d matrices of Ref. [31], whose elements are independent integer random numbers chose uniformly in $\{1, 2, \ldots, N^2\}$. These matrices are described in Sec. VI A. We ran simulations on such cost matrices of sizes N ranging in size between 128 and 8192 for the CPU-based programs, and between 128 and 32 768 for the GPU-based programs, with two exceptions. As LAP has a time complexity of $O(N^4)$, we limited its application to matrices up to size 4096. Second, the available version of HunCUDA is limited to $N \leqslant 8192$. For each value of N, 10 simulations were performed. The average computing times over the 10 simulations for the different programs are plotted against this size N in Figs. 5(a) and 5(b). The corresponding

TABLE I. Time complexities of different algorithms for solving integer assignment problems.

Processor	Method	Worst case complexity[a]	Random matrices[b]	Machol-Wien matrices[c]
CPU	LAP [35]	$O(N^4)$	3.7[d]	4.0
CPU	LAPJV [37]	$O(N^3)$	2.2	2.9
CPU	SKAP [39]	$O(N^3)$	2.8	2.9
CPU	UMatching	$O(N^3)$	1.6	1.4
GPU	HunCUDA [41]	$O(N^3)$	1.9	2.8
GPU	RAPIDS [43]	$O(N^3)$	1.4	2.2
GPU	UMatching	$O(N^3)$	1.4	1.3

[a]Worst case complexity of the corresponding algorithm. See text for details.
[b]Cost matrices include elements are an independent integer random numbers chose uniformly in $\{1, 2, \ldots, N^2\}$, see Ref. [31].
[c]Cost matrix C defined as $C(i, j) = (i - 1) \times (j - 1)$ for $(i, j) \in [1, N]^2$.
[d]Time complexity, as evaluated from of linear fit of the log of the computing time verse the log of the size of the problem (see Fig. 5).

apparent time complexities for all programs that are tested are given in Table I.

For the CPU-based programs, setting aside small values of N (i.e., below 1000) for which initialization costs dominate, we observe that, as expected, LAPJV and SKAP are significantly faster than LAP and UMatching. LAP has a worst case complexity of $O(N^4)$; its apparent complexity on the random integer cost matrices is close to this worst case. LAPJV and SKAP, both based on improving a matching along augmenting paths, are fast, with observed time complexity better than 3. Computing times for UMatching are found to be between those of LAPJV/SKAP and those of LAP. Its interest, however, comes in its apparent time complexity, 1.6 (see Table I), i.e., significantly better than those of LAP, LAPJV, and SKAP. We first note that our implementation relies heavily on linear algebra, as at each inverse temperature we solve a non linear system of equation iteratively. Each step requires solving a linear system of equations, for which we use an iterative procedure (see details in the implementation section). We have relied on the optimized BLAS and LAPACK libraries for all those operations involving linear algebra. Those libraries are optimized for the processors we use and make efficient use of the multiple cores available. Such operations would therefore benefit from even more parallelization available for example on the GPU.

UMatching is found to be between 2 and 4 times slower that Hungarian RAPIDS on the random integer assignment problems, but faster than HunCUDA for $N \geqslant 1000$. We note that HunCUDA is based on the $O(N^4)$ version of the Hungarian algorithm (see Ref. [41]) and as such is expected to underperform for large N. Both RAPIDS and UMatching have similar apparent time complexities, $O(N^{1.4})$ (see Table I).

Our second benchmark test involves the Machol-Wien cost matrices. As described above, those matrices are considered hard for the Hungarian algorithm, both for its matrix-based implementation and its graph-based implementation. We ran similar tests as those described above for random integer cost matrices, for all the CPU-based and GPU-based programs considered. The mean computing times over 10 simulations for the different programs are plotted against the size N of the square cost matrix in Figs. 5(c) and 5(d). The corresponding apparent time complexities for all programs are given in Table I.

As expected, LAP has an apparent time complexity of $O(N^4)$ and it is the slowest of the 4 CPU-based program on those Machol-Wien cost matrices. LAPJV and SKAP also have apparent time complexities close to their worst case values. In addition, all three programs run significantly slower on those matrices than on the random integer cost matrices. In contrast, UMatching is found to run faster on the Machol-Wien matrices than on the random inter cost matrices (by close to a factor of 3), with a similar apparent time complexity ($O(N^{1.4})$ for Machol-Wien matrices, compared to $O(N^{1.6})$ for random integer matrices). IUMatching CPU is found to be faster than LAPJV and SKAP for matrices of size 8192. The difference is even more striking when we consider the GPU-based programs. UMatching GPU is found to be significantly faster on the Machol-Wien matrices than all GPU implementations of the Hungarian algorithm we have tested, including the fast implementation available in the RAPIDS suite. Compared to the CPU case, UMatching computing times on the GPU do not differ significantly between the random integer cost matrices and the Machol-Wien matrices. While we cannot fully exclude that there could some types of matrices that could lead to worst-case scenario for UMatching, we believe that differences in computing time will remain small for all types of matrices.

2. Comparing UMatching and SKAP for genuine k-cardinality assignment problems

In the previous subsection, UMatching was compared with fast CPU and GPU implementations of the Hungarian algorithm on different types of balanced assignment problems. We further assessed UMatching on explicit k-cardinality assignment problems. Of all programs considered above, only SKAP can solve genuine k-cardinality problems. We compared it with the CPU and the GPU versions of UMatching. We considered both integer random matrices and the specific case of the Machol-Wien matrices. Results are presented in Fig. 6.

We considered first random integer cost matrices whose elements are independent integer random numbers chose uniformly in $\{1, 2, \ldots, N^2\}$ of size 8000×8000, for which we solve the k-cardinality assignment problem for values of k varying from 1000 to 8000. The expected minimal cost of such a k-cardinality assignment is given by Eq. (23). For each

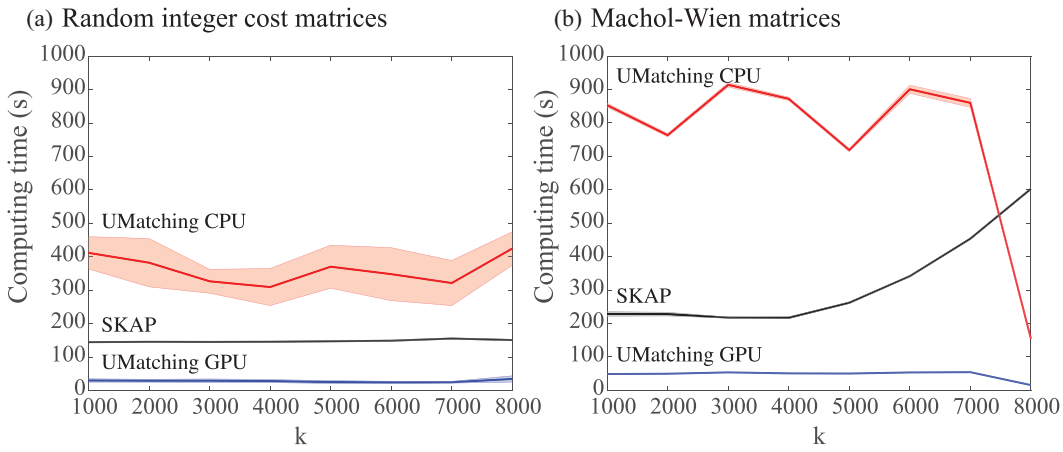

FIG. 6. The computing time T as a function of k when solving a k-cardinality assignment problem with either an integer random cost matrix C of size $N \times N$ with $N = 8000$ whose elements are independent integer random numbers chose uniformly in $\{1, 2, \ldots, N^2\}$ (a), or with a Machol-Wien cost matrix (see text for details) of the same size (b). At each value of k, 10 experiments were performed; the mean values are shown as a solid line and the corresponding standard deviations are illustrated with a shaded areas. All computations were run on a computer with an AMD Ryzen Threadripper PRO 3975WX with 32 CPUs and a NVIDIA RTX A5000.

value of k, we ran 10 different random instances and computed the mean value and standard deviation of the corresponding computing time at convergence. UMatching CPU is found to be 2.5 times slower than SKAP, while UMatching GPU is found to be 5 times faster than SKAP. As observed in Fig. 6(a), the computing times associated with both versions of UMatching (CPU and GPU), as well as the computing times associated with SKAP are relatively constant with respect to k. While in the original paper describing SKAP [39] there were no mentions of time complexity, experimental evidence showed that its computing time is relatively independent of k, at least for random matrices (Table 4 in Ref. [39]). For UMatching, the dependency with respect to k is more complex. Note first that the number of unknowns in the SPA system of equations is independent of k, and therefore the computing time for solving this system is expected to be also independent of k. However, as the annealing procedure in function of the inverse temperature β is stopped when the transport plan $\overline{G}_\beta$ contains exactly k values that are greater or equal to $\frac{1}{2}$, a dependency of the computing time with k is expected. This dependency, however, is observed to be weak, at least for random integer cost matrices.

We then considered Wachol-Wien matrices of size 8000×8000, for which we solve the k-cardinality assignment problem for values of k varying from 1000 to 8000. For each value of k, we ran 10 different optimizations and computed the mean value and standard deviation of the corresponding computing time at convergence. Note that as this problem is deterministic, fluctuations over the different runs are expected to be small. A comparison of Figs. 6(a) and 6(b) shows that all the computing times of SKAP, UMatching CPU, and UMatching GPU differ significantly between the two types of matrices. While SKAP's computing times are relatively constant with respect to k for random integer matrices, they increase as a function of k for the Machol-Wien matrices. In contrast, both versions of UMatching show relatively constant computing times for $k < N$, and a significant drop when $k = N$. We assign this behavior to the fact that the k-cardinality assignment problem for Machol-Wien matrices is degenerate for $k < N$ while it has a unique solution when $k = N$. For $k < N$, UMatching CPU is found to be 5.5 times slower than SKAP, while UMatching GPU is found to be 5.5 times faster than SKAP. For $k = N$, however, both UMatching CPU and UMatching GPU are found to be faster than SKAP, by factors of 4 and 40, respectively.

VII. CONCLUSION

In this article, we developed a statistical physics framework to solve the unbalanced k-cardinality assignment problem. Given two sets of points S_1 and S_2 with possibly different cardinalities N_1 and N_2, a cost matrix for assignments between points of those sets, and an imposed number of assignments k, we have constructed a free energy parametrized by temperature that captures the constraints of the k-assignment problem. This free energy is concave, and its maximum defines an optimal k-assignment between the two sets of points. We proved that it decreases monotonically as a function of the inverse of temperature to the optimal assignment cost, lending itself to temperature annealing. We also proved that for small enough temperatures, the exact solution to the generic k-assignment problem can be derived directly by simply rounding off to the nearest integer the elements of the computed assignment matrix. We have also provided a provably convergent method to handle degenerate k-assignment problems. We have described two implementations of our methods that are optimized for the CPU and GPU parallel architectures, respectively. We have shown that the latter is competitive with state-of-the-art parallel codes that implement the Hungarian algorithms for generic assignment problems, and significantly faster (orders of magnitude) than those implementations for hard assignment problems.

The framework we have proposed can be applied to balanced as well as to k-cardinality assignment problems. In its formulation introduced in this paper, it suffers the same limitations as the Hungarian problem and cannot be directly applied to augmented assignment problems in which multiple

tasks can be assigned to a single agent, or in which a single task requests multiple agents. Analyses of those augmented assignment problems will be the topic of a future paper.

ACKNOWLEDGMENTS

The work discussed here originated from a visit by P.K. to the Institut de Physique Théorique, French Alternative Energies and Atomic Energy Commission (CEA) Saclay, France. He thanks them for their hospitality and financial support.

APPENDIX A: REMINDER ON DOUBLY STOCHASTIC AND DOUBLY SUBSTOCHASTIC MATRICES

An $N \times N$ matrix $A = (a(i, j))$ is said to be doubly stochastic if and only if it satisfies the following conditions:

$$a(i, j) \geqslant 0,$$
$$\sum_{i=1}^{N} a(i, j) = 1, \quad \sum_{j=1}^{N} a(i, j) = 1,$$

for all $(i, j) \in [1, N]^2$. The set of doubly stochastic matrices of size $N \times N$ is a convex polytope whose vertices are the permutation matrices with the same size. This is expressed by the following theorem, established by Birkhoff [9] and von Neumann [10]:

Theorem 8. An $N \times N$ matrix A is doubly stochastic if and only if it can be written as a weighted sum of permutation matrices, i.e.,

$$A = \sum_{\pi \in \Pi_N} a_\pi \pi,$$

where Π_N is the set of permutation matrices of size N, a_π is a positive real number and $\sum_{\pi \in \Pi_N} a_\pi = 1$.

Doubly stochastic matrices and their properties associated to the Birkhoff–von Neumann theorem above proved useful to establish convergence properties of statistical physics frameworks for solving the balanced assignment problem [16,45].

For the unbalanced k-cardinality assignment problem, however, we need to consider a different, although related set of matrices, those that are doubly substochastic. A $N_1 \times N_2$ matrix $A = (a(i, j))$ is double substochastic if it satisfies

$$a(i, j) \geqslant 0,$$
$$\sum_{i=1}^{N} a(i, j) \leqslant 1, \quad \sum_{j=1}^{N} a(i, j) \leqslant 1,$$

for all $(i, j) \in [1, N_1] \times [1, N_2]$. The set of doubly substochastic matrices of size $N_1 \times N_2$ is also a convex polytope whose vertices are the partial permutation matrices with the same size [46,47].

Our need is in fact even more specific as it relies on the subset of doubly substochastic matrices with a fixed total sum of all their elements. Let us define

$$\sigma(A) = \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} a(i, j).$$

A partial permutation matrix π of size $N_1 \times N_2$ and of rank k is a doubly substochastic matrix of size $N_1 \times N_2$ whose elements belong to $\{0, 1\}$, with no more than one 1 on any row or any column, and with $\sigma(\pi) = k$. We will write $P_{N_1,N_2}(k)$ the set of such partial permutation matrix. A matrix π of this set can be wcharacterized with the following injection f_π:

$$f_\pi : [1, N_1] \to \{0\} \cup [1, N_2],$$
$$i \mapsto f_\pi(i) = \begin{cases} 0 & \text{if } \sum_{j=1}^{N_2} \pi(i, j) = 0, \\ j & \text{if } \pi(i, j) = 1. \end{cases} \tag{A1}$$

The set of doubly substochastic matrix A with $\sigma(A) = k$, k fixed, satisfies a property similar to the Birkhoff–von Neumann theorem, as established by Mendelsohn and Dulmage for square matrices [48], and later by Brualdi and Lee for rectangular matrices [49]:

Theorem 9. An $N_1 \times N_2$ doubly substochastic matrix A with $\sigma(A) = k$ can be written as a weighted sum of partial permutation matrices of rank k, i.e.,

$$A = \sum_{\pi \in P_{N_1,N_2}(k)} a_\pi \pi,$$

where a_π is a positive real number and $\sum_{\pi \in P_{N_1,N_2}(k)} a_\pi = 1$.

Finally, we note that any doubly substochastic matrix A of size $N_1 \times N_2$ can be augmented to a doubly stochastic matrix A^a of size $N_1 + N_2$:

$$A^a = \begin{bmatrix} A & I_{N_1} - D_1 \\ I_{N_2} - D_2 & A^T \end{bmatrix}, \tag{A2}$$

where I_N is the identify matrix of size $N \times N$, D_1 and D_2 are the diagonal matrices containing the row sums and column sums of the matrix A:

$$D_1(i, i) = \sum_j A(i, j),$$
$$D_2(j, j) = \sum_i A(i, j).$$

APPENDIX B: PROOF OF THEOREM 1: CONCAVITY OF THE EFFECTIVE FREE ENERGY

We first prove that the effective free energy $F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)$ is weakly concave, by showing that its Hessian H is negative definite. H is a symmetric matrix of size $(N_1 + N_2 + 1) \times (N_1 + N_2 + 1)$, such that its rows and columns correspond to all N_1 λ values first, followed by all N_2 μ values, and finally to the value x. Let h' be the derivative of the function $h(x) = 1/(1 + e^x)$, i.e.,

$$h'(x) = -\frac{e^x}{(1 + e^x)^2}. \tag{B1}$$

We note first that $h'(x) \in [\frac{-1}{4}, 0) \quad \forall x \in \mathbb{R}$, i.e., that $h'(x)$ is always strictly negative. We define the matrix X' and the vector $\mathbf{d_1'}$ and $\mathbf{d_2'}$ such that

$$X'(i,j) = h'(\beta(C(i,j) + \lambda(i) + \mu(j) + x), \quad d_1'(i) = h'(-\beta\lambda(i)), \quad d_2'(j) = h'(-\beta\mu(j)).$$

From Eqs. (10), we obtain

$$\begin{aligned}
H(i,i') &= \frac{\partial^2 F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)}{\partial\lambda(i)\partial\lambda(i')} = \beta\delta_{ii'}\left(\sum_j X'(i,j) + d_1'(i)\right),\\
H(i,j) &= \frac{\partial^2 F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)}{\partial\lambda(i)\partial\mu(j)} = \beta X'(i,j),\\
H(i,N) &= \frac{\partial^2 F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)}{\partial\lambda(i)\partial x} = \beta\sum_j X'(i,j),\\
H(j,j') &= \frac{\partial^2 F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu})}{\partial\mu(j)\partial\mu(j')} = \beta\delta_{jj'}\left(\sum_i X'(i,j) + d_2'(j)\right),\\
H(j,N) &= \frac{\partial^2 F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)}{\partial\mu(j)\partial x} = \beta\sum_i X'(i,j),\\
H(N,N) &= \frac{\partial^2 F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)}{\partial x^2} = \beta\sum_i\sum_j X'(i,j),
\end{aligned}$$

where δ are Kronecker functions, the indices i and i' belong to $[1, N_1]$ and the indices j and j' belong to $[1, N_2]$, and we have defined $N = N_1 + N_2 + 1$.

Let $\mathbf{x} = (\mathbf{x_1}, \mathbf{x_2}, x_3)$ be an arbitrary vector of size N. The quadratic form $Q(\mathbf{x}) = \mathbf{x}^T H\mathbf{x}$ is equal to

$$\begin{aligned}
Q(\mathbf{x}) &= \sum_{i,i'} x_1(i)H(i,i')x_1(i') + \sum_{j,j'} x_2(j)H(j,j')x_2(j') + x_3H(N,N)x_3\\
&\quad + 2\sum_{i,j} x_1(i)H(i,j)x_2(j) + 2\sum_i x_1(i)H(i,N)x_3 + 2\sum_j x_2(j)H(j,N)x_3\\
&= \beta\sum_{i,j}(x_1(i) + x_2(j) + x_3)^2 X'(i,j) + \beta\left(\sum_i x_1(i)^2 d_1'(i) + \sum_j x_2(j)^2 d_2'(j)\right).
\end{aligned}$$

As $X'(i,j)$, $d_1'(i)$, and $d_2'(j)$ are all based on the function h' that is strictly negative, the summands in the equation above are negative for all $i \in [1, N_1]$ and $j \in [1, N_2]$, and therefore $Q(\mathbf{x})$ is negative for all vector $\mathbf{x}$. The Hessian H is negative, semidefinite.

As $Q(\mathbf{x})$ is a sum of negative terms, it is 0 if and only if all the terms are equal to 0. As the function $h'(x)$ is strictly negative, this means that $\forall(i,j)$,

$$(x_1(i) + x_2(j) + x_3)^2 = 0, \quad x_1(i)^2 = 0, \quad x_2(j)^2 = 0.$$

This is realized when all $x_1(i)$, $x_2(j)$, and when x_3 are zero, namely thet $\mathbf{x} = \mathbf{0}$. Therefore, H is negative, definite, and the free energy is strictly concave.

APPENDIX C: PROOF OF PROPOSITION 1: RETRIEVING THE TRANSPORTATION PLAN AND THE INDICATOR FUNCTIONS FROM THE SPA SOLUTIONS

In the unbalanced k-cardinality assignment problem between two sets S_1 and S_2, the "system" is characterized with a partition function [Eq. (6)],

$$\begin{aligned}
Z(\beta) = &\sum_{G(i,j)\in\{0,1\}}\sum_{n_1(i)\in\{0,1\}}\sum_{n_2(j)\in\{0,1\}} e^{-\beta\sum_{i,j}C(i,j)G(i,j)}\prod_i\delta\left(\sum_j G(i,j) - n_1(i)\right)\\
&\times\prod_j\delta\left(\sum_i G(i,j) - n_2(j)\right)\delta\left(\sum_{i,j} G(i,j) - k\right),
\end{aligned}$$

with a corresponding effective free energy,

$$F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x) = -\frac{1}{\beta}\sum_i \ln\left(1 + e^{\beta\lambda(i)}\right) - \frac{1}{\beta}\sum_j \ln\left(1 + e^{\beta\mu(j)}\right) - \frac{1}{\beta}\sum_{i,j} \ln\left(1 + e^{-\beta(C(i,j)+\lambda(i)+\mu(j)+x)}\right) - kx.$$

F_β is a function of $N_1 + N_2 + 1$ variables, namely $\lambda(i)$ for $i \in [1, N_1]$, $\mu(j)$ for $j \in [1, N_2]$, and x. The values of these variables that solve the SPA conditions are referred to as $\lambda^{\mathrm{MF}}(i)$, $\mu^{\mathrm{MF}}(j)$, and x^{MF}, respectively. Those values define $X^{\mathrm{MF}}(i, j)$, $d_1^{\mathrm{MF}}(i)$, and $d_2^{\mathrm{MF}}(j)$, as given in Eq. (11). We show that those values are the solutions to the unbalanced k-cardinality assignment problem.

To find those solutions, namely the expected values $\overline{G}(i, j)$, $\overline{n}_1(i)$, and $\overline{n}_2(j)$ for the transportation plan, and indicator functions of the elements of S_1 and S_2 that are in correspondence, respectively, we need to introduce three vector fields $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w}$, and a variable s and modify the partition function:

$$Z(\beta) = \sum_{G(i,j)\in\{0,1\}} \sum_{n_1(i)\in\{0,1\}} \sum_{n_2(j)\in\{0,1\}} e^{-\beta\sum_{i,j} C(i,j)G(i,j)} e^{\beta\left(\sum_{i,j} G(i,j)u(i,j) + \sum_i v(i)n_1(i) + \sum_j w(j)n_2((j)+xs\right)}$$
$$\times \prod_i \delta\left(\sum_j G(i, j) - n_1(i)\right) \prod_j \delta\left(\sum_i G(i, j) - n_2(j)\right) \delta\left(\sum_{i,j} G(i, j) - k\right).$$

Following the same procedure as described in the main text for evaluating this modified partition function, we find

$$F = F_\beta(\mathbf{u}, \mathbf{v}, \mathbf{w}, s) = -\frac{1}{\beta}\sum_i \ln[1 + e^{\beta(\lambda(i)+v(i))}] - \frac{1}{\beta}\sum_j \ln[1 + e^{\beta(\mu(j)+w(j))}]$$
$$- \frac{1}{\beta}\sum_{i,j} \ln[1 + e^{-\beta(C(i,j)+\lambda(i)+\mu(j)+x-u(i,j))}] - kx - sx.$$

The expected transportation matrix $\overline{G}(i, j)$ between point i in S_1 and point j in S_2 is given by

$$\overline{G}(i, j) = -\frac{\partial F}{\partial u(i, j)}\bigg|_{\mathbf{u}=\mathbf{0},\mathbf{v}=\mathbf{0},\mathbf{w}=\mathbf{0},s=0,\boldsymbol{\lambda}=\boldsymbol{\lambda}^{\mathrm{MF}},\boldsymbol{\mu}=\boldsymbol{\mu}^{\mathrm{MF}},x=x^{\mathrm{MF}}},$$

i.e.,

$$\overline{G}(i, j) = \frac{1}{1 + e^{\beta(C(i,j)+\lambda^{\mathrm{MF}}(i)+\mu^{\mathrm{MF}}(j)+x^{\mathrm{MF}})}} = h(\beta(C(i, j) + \lambda^{\mathrm{MF}}(i) + \mu^{\mathrm{MF}}(j) + x^{\mathrm{MF}})) = X^{\mathrm{MF}}(i, j).$$

Similarly, the expected indicator function $\overline{n}_1(i)$ over S_1 are

$$\overline{n}_1(i) = -\frac{\partial F}{\partial v(i)}\bigg|_{\mathbf{u}=\mathbf{0},\mathbf{v}=\mathbf{0},\mathbf{w}=\mathbf{0},s=0,\boldsymbol{\lambda}=\boldsymbol{\lambda}^{\mathrm{MF}},\boldsymbol{\mu}=\boldsymbol{\mu}^{\mathrm{MF}},x=x^{\mathrm{MF}}},$$
$$\overline{n}_2(j) = -\frac{\partial F}{\partial w(j)}\bigg|_{\mathbf{u}=\mathbf{0},\mathbf{v}=\mathbf{0},\mathbf{w}=\mathbf{0},s=0,\boldsymbol{\lambda}=\boldsymbol{\lambda}^{\mathrm{MF}},\boldsymbol{\mu}=\boldsymbol{\mu}^{\mathrm{MF}},x=x^{\mathrm{MF}}},$$

i.e.,

$$\overline{n}_1(i) = \frac{1}{1 + e^{-\beta\lambda^{\mathrm{MF}}(i)}} = h(-\beta\lambda^{\mathrm{MF}}(i)) = d_1^{\mathrm{MF}}(i),$$
$$\overline{n}_2(j) = \frac{1}{1 + e^{-\beta\mu^{\mathrm{MF}}(j)}} = h(-\beta\mu^{\mathrm{MF}}(j)) = d_2^{\mathrm{MF}}(j).$$

Finally,

$$\overline{x} = -\frac{\partial F}{\partial s}\bigg|_{\mathbf{u}=\mathbf{0},\mathbf{v}=\mathbf{0},\mathbf{w}=\mathbf{0},\boldsymbol{\lambda}=\boldsymbol{\lambda}^{\mathrm{MF}},\boldsymbol{\mu}=\boldsymbol{\mu}^{\mathrm{MF}},x=x^{\mathrm{MF}}},$$

i.e.,

$$\overline{x} = x^{\mathrm{MF}},$$

which concludes the proof of Proposition 1.

APPENDIX D: MONOTONICITY OF F_β^{MF}

The effective free energy $F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)$ defined in Eq. (8) is a function of the cost matrix C and of real unconstrained variables $\lambda(i)$, $\mu(j)$, and x. For the sake of simplicity, for any $(i, j) \in [1, N_1] \times [1, N_2]$, we define

$$y(i, j) = C(i, j) + \lambda(i) + \mu(j) + x, \quad y^{\mathrm{MF}}(i, j) = C(i, j) + \lambda^{\mathrm{MF}}(i) + \mu^{\mathrm{MF}}(j) + x^{\mathrm{MF}}.$$

The effective free energy is then

$$F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x) = -\frac{1}{\beta}\sum_i \ln\left(1 + e^{\beta\lambda(i)}\right) - \frac{1}{\beta}\sum_j \ln\left(1 + e^{\beta\mu(j)}\right) - \frac{1}{\beta}\sum_{i,j} \ln\left(1 + e^{-\beta y(i,j)}\right) - kx. \tag{D1}$$

As written above, $F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)$ is a function of the independent variables β, $\lambda(i)$, $\mu(j)$, and x. However, under the saddle point approximation, the variables $\lambda(i)$, $\mu(j)$, and x are constrained by the conditions

$$\frac{\partial F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)}{\partial \lambda(i)} = 0, \quad \frac{\partial F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)}{\partial \mu(j)} = 0, \quad \frac{\partial F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)}{\partial x} = 0, \tag{D2}$$

and the free energy under those constraints is written as F_β^{MF}. In the following, we will use the notations $\frac{dF_\beta^{\mathrm{MF}}}{d\beta}$ and $\frac{\partial F_\beta^{\mathrm{MF}}}{\partial\beta}$ to differentiate between the total derivative and partial derivative of F_β^{MF} with respect to β, respectively. Based on the chain rule,

$$\frac{dF_\beta^{\mathrm{MF}}}{d\beta} = \frac{\partial F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)}{\partial\beta} + \sum_i \frac{\partial F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)}{\partial\lambda(i)}\frac{\partial\lambda(i)}{\partial\beta} + \sum_l \frac{\partial F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)}{\partial\mu(j)}\frac{\partial\mu(j)}{\partial\beta} + \frac{\partial F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)}{\partial x}\frac{\partial x}{\partial\beta}.$$

Using the constraints (D2), we find that

$$\frac{dF_\beta^{\mathrm{MF}}}{d\beta} = \frac{\partial F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x)}{\partial\beta},$$

namely that the total derivative with respect to β is in this specific case equal to the corresponding partial derivative, which is easily computed to be

$$\begin{aligned}\frac{dF_\beta^{\mathrm{MF}}}{d\beta} = {} & \frac{1}{\beta^2}\sum_i \left(\ln(1 + e^{\beta\lambda^{\mathrm{MF}}(i)}) - \frac{\beta\lambda^{\mathrm{MF}}(i)}{1 + e^{-\beta\lambda^{\mathrm{MF}}(i)}}\right) + \frac{1}{\beta^2}\sum_j \left(\ln(1 + e^{\beta\mu^{\mathrm{MF}}(j)}) - \frac{\beta\mu^{\mathrm{MF}}(j)}{1 + e^{-\beta\mu^{\mathrm{MF}}(j)}}\right) \\ & + \frac{1}{\beta^2}\sum_{i,j} \left(\ln(1 + e^{-\beta y^{\mathrm{MF}}(i,j)}) + \frac{\beta y^{\mathrm{MF}}(i, j)}{1 + e^{\beta y^{\mathrm{MF}}(i,j)}}\right).\end{aligned} \tag{D3}$$

Let $t(x) = \ln(1 + e^{-x}) + \frac{x}{1+e^x}$. The function $t(x)$ is continuous and defined over all real values x and is bounded below by 0, i.e., $t(x) \geqslant 0 \quad \forall x \in \mathbb{R}$. As

$$\frac{dF_\beta^{\mathrm{MF}}}{d\beta} = \frac{1}{\beta^2}\left(\sum_i t(-\beta\lambda(i)) + \sum_j t(-\beta\mu(j)) + \sum_{i,j} t(\beta y(i, j)\right),$$

we conclude that

$$\frac{dF_\beta^{\mathrm{MF}}}{d\beta} \geqslant 0,$$

namely that F_β^{MF} is a monotonically increasing function of β.

APPENDIX E: MONOTONICITY OF U_β^{MF}

Let

$$U_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x) = \sum_{i,j} C(i, j)\bar{G}(i, j), \tag{E1}$$

and let the corresponding mean-field approximation of the internal energy at the saddle point

$$U_\beta^{\rm MF} = U_\beta(\boldsymbol{\lambda}^{\rm MF}, \boldsymbol{\mu}^{\rm MF}, x^{\rm MF}). \tag{E2}$$

Before computing $\frac{dU_\beta^{\rm MF}}{d\beta}$, we prove the following property.

Property 4.

$$U_\beta^{MF} = F_\beta^{MF} + \beta \frac{dF_\beta^{MF}}{d\beta}, \tag{E3}$$

i.e., it extends the well known relationship between the free energy and the average energy to their mean-field counterparts.

Proof. Using Eqs. (D1), (D3), (13), and the definition of $h(x) = 1/(1+e^x)$, we find that

$$\beta \frac{dF_\beta^{\rm MF}}{d\beta} = -F_\beta^{\rm MF} - kx^{\rm MF} - \sum_i \lambda^{\rm MF}(i)\overline{n}_{1\beta}(i) - \sum_j \mu^{\rm MF}(j)\overline{n}_{2\beta}(j) + \sum_{i,j} y^{\rm MF}(i,j)\overline{G}(i,j).$$

Let us recall that

$$y^{\rm MF}(i,j) = C(i,j) + \lambda(i)^{\rm MF} + \mu(l)^{\rm MF} + x^{\rm MF}.$$

In addition, all mean-field values correspond to the maximum of the effective free energy, for which the constraints are satisfied, namely $\sum_j \overline{G}(i,j) = \overline{n}_{1\beta}(i)$ and $\sum_i \overline{G}(i,j) = \overline{n}_{2\beta}(j)$. Replacing in Eq. (E4), we get

$$\beta \frac{dF_\beta^{\rm MF}}{d\beta} = -F_\beta^{\rm MF} - kx^{\rm MF} - \sum_{i,j} \lambda^{\rm MF}(i)\overline{n}_{1\beta}(i) - \sum_{i,j} \mu^{\rm MF}(j)\overline{n}_{2\beta}(j) + \sum_{i,j} (C(i,j) + \lambda^{\rm MF}(i) + \mu^{\rm MF}(j) + x^{\rm MF})\overline{G}(i,j),$$

i.e.,

$$\beta \frac{dF_\beta^{\rm MF}}{d\beta} = -F_\beta^{\rm MF} + \sum_{i,j} C(i,j)\overline{G}(i,j) = -F_\beta^{\rm MF} + U_\beta^{\rm MF}, \tag{E4}$$

which concludes the proof. ■

Based on the chain rule,

$$\frac{dU_\beta^{\rm MF}}{d\beta} = \frac{\partial U_\beta^{\rm MF}}{\partial \beta} + \sum_i \frac{\partial U_\beta^{\rm MF}}{\partial \lambda(i)}\frac{\partial \lambda(i)}{\partial \beta} + \sum_j \frac{\partial U_\beta^{\rm MF}}{\partial \mu(j)}\frac{\partial \mu(j)}{\partial \beta} + \frac{\partial U_\beta^{\rm MF}}{\partial x}\frac{\partial x}{\partial x}.$$

Let us compute all partial derivatives in this equation using the Property 4 proved above:

$$\frac{\partial U_\beta^{\rm MF}}{\partial \lambda(i)} = \frac{\partial F_\beta^{\rm MF}}{\partial \lambda(i)} + \beta \frac{\partial}{\partial \lambda(i)}\left(\frac{\partial F_\beta^{\rm MF}}{\partial \beta}\right) = \frac{\partial F_\beta^{\rm MF}}{\partial \lambda(i)} + \beta \frac{\partial}{\partial \beta}\left(\frac{\partial F_\beta^{\rm MF}}{\partial \lambda(i)}\right) = 0,$$

where the zero is a consequence of the SPA constraints. Similarly, we find

$$\frac{\partial U_\beta^{\rm MF}}{\partial \mu(j)} = 0, \quad \frac{\partial U_\beta^{\rm MF}}{\partial x} = 0.$$

Finally,

$$\frac{\partial U_\beta^{\rm MF}}{\partial \beta} = 2\frac{\partial F_\beta^{\rm MF}}{\partial \beta} + \beta \frac{\partial}{\partial \beta}\left(\frac{\partial F_\beta^{\rm MF}}{\partial \beta}\right) = 2\frac{\partial F_\beta^{\rm MF}}{\partial \beta} + \beta\left(-\frac{2}{\beta}\frac{\partial F_\beta^{\rm MF}}{\partial \beta}\right) + \sum_i (-\lambda(i)t'(-\beta\lambda^{\rm MF}(i))$$
$$+ \sum_j (-\mu^{\rm MF}(j)t'(-\beta\mu^{\rm MF}(j)) + \sum_{ki,j} y^{\rm MF}(i,j)t'(\beta y^{\rm MF}(i,j)),$$

where $t(x)$ is the function define above, and $t'(x)$ its derivative: $t'(x) = -\frac{xe^x}{(1+e^x)^2}$. Let us define $g(x) = xt'(x)$; $g(x)$ is negative, bounded above by 0. Then,

$$\frac{\partial U_\beta^{\rm MF}}{\partial \beta} = \frac{1}{\beta}\sum_i g(-\beta\lambda^{\rm MF}(i)) + \frac{1}{\beta}\sum_j g(-\beta\mu^{\rm MF}(j)) + \frac{1}{\beta}\sum_{i,j} g(\beta y^{\rm MF}(i,j)).$$

Therefore,

$$\frac{dU_\beta^{\rm MF}}{d\beta} = \frac{\partial U_\beta^{\rm MF}}{\partial \beta} \leqslant 0,$$

and the function $U_\beta^{\rm MF}$ is a monotonically decreasing function of β.

APPENDIX F: REWRITING THE EFFECTIVE FREE ENERGY

Recall that we have defined $h(x) = \frac{1}{1+e^x}$ for $x \in \mathbb{R}$ and $f(x) = -(1-x)\ln(1-x) - x\ln(x)$, for $x \in (0, 1)$. Let us define $t(x) = \ln(1+e^{-x}) + xh(x)$. We establish first the following identity.

Property 5.

$$t(x) = \ln(1+e^{-x}) + xh(x) = -(1-h(x))\ln(1-h(x)) - h(x)\ln(h(x)) = f(h(x)). \tag{F1}$$

Proof. Let x be a real number and let $y = h(x)$. We know that $y \in (0, 1)$. We rewrite $1 + e^{-x}$ and x as functions of y:

$$1 + e^{-x} = \frac{1}{1-y} \quad x = \ln(1-y) - \ln(y).$$

Then,

$$t(x) = \ln\left(\frac{1}{1-y}\right) + \ln(1-y)y - \ln(y)y = -(1-y)\ln(1-y) - y\ln y,$$

which concludes the proof, as $y = h(x)$. ■

Recall now that we have defined

$$y(i,j) = C(i,j) + \lambda(i) + \mu(j) + x, \quad X(i,j) = h(\beta y(i,j)), \quad d_1(i) = h(-\beta\lambda(i)), \quad d_2(j) = h(-\beta\mu(j)).$$

Let us now rewrite the free energy,

$$F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x) = -\frac{1}{\beta}\sum_i \ln\left(1 + e^{\beta\lambda(i)}\right) - \frac{1}{\beta}\sum_j \ln\left(1 + e^{\beta\mu(j)}\right) - \frac{1}{\beta}\sum_{i,j} \ln\left(1 + e^{-\beta y(i,j)}\right) - kx.$$

The internal energy is

$$U_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x) = \sum_{i,j} C(i,j)X(i,j).$$

Adding, and subsequently subtracting the internal energy in the equation defining the free energy, we get

$$F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x) = U_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x) - \sum_{i,j} C(i,j)h(\beta y(i,j)) - \frac{1}{\beta}\sum_i \ln[1 + e^{\beta\lambda(i)}] - \frac{1}{\beta}\sum_j \ln[1 + e^{\beta\mu(j)}] - \frac{1}{\beta}\sum_{i,j} \ln[1 + e^{-\beta y(i,j)}] - kx.$$

Replacing $C(i,j)$ with $y(i,j) - \lambda(i) - \mu(j) - x$, we get (after reorganization)

$$\begin{aligned} F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x) = U_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x) &- \frac{1}{\beta}\sum_i (\ln(1 + e^{\beta\lambda(i)}) + \beta\lambda(i)h(-\beta\lambda(i))) - \frac{1}{\beta}\sum_j (\ln(1 + e^{\beta\mu(j)}) + \beta\mu(j)h(-\beta\mu(j))) \\ &- \frac{1}{\beta}\sum_{i,j} (\ln(1 + e^{-\beta y(i,j)}) + \beta y(i,j)h(\beta y(i,j))) + x\left(\sum_{i,j} X(i,j) - k\right) \\ &+ \sum_i \lambda(i)\left(\sum_j X(i,j) - d_1(i)\right) + \sum_j \mu(j)\left(\sum_i X(i,j) - d_2(j)\right). \end{aligned}$$

Therefore, using Property 5 above,

$$\begin{aligned} F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x) = U_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x) &- \frac{1}{\beta}\sum_i f(h(-\beta\lambda(i))) - \frac{1}{\beta}\sum_j f(h(-\beta\mu(j))) - \frac{1}{\beta}\sum_{i,j} f(h(\beta y(i,j))) + x\left(\sum_{i,j} X(i,j) - k\right) \\ &+ \sum_i \lambda(i)\left(\sum_j X(i,j) - d_1(i)\right) + \sum_j \mu(j)\left(\sum_i X(i,j) - d_2(j)\right), \end{aligned}$$

which concludes the proof of Eq. (15), taking into account that $T = \frac{1}{\beta}$.

APPENDIX G: PROOF OF THEOREM 3: CONVERGENCE OF THE MEAN-FIELD FREE ENERGY AND THE INTERNAL ENERGY TO THE OPTIMAL k-ASSIGNMENT COST

For simplicity in notation, we define $F^{\mathrm{MF}}(\infty) = \lim_{\beta\to+\infty} F_\beta^{\mathrm{MF}}$ and $U^{\mathrm{MF}}(\infty) = \lim_{\beta\to+\infty} U_\beta^{\mathrm{MF}}$.

1. Proof that $F^{\mathrm{MF}}(\infty) = U^{\mathrm{MF}}(\infty)$

Proof. The mean-field free energy and internal energy are related by Eqs. (17). Based on these equations, the mean-field entropy can be written as

$$S_\beta^{\mathrm{MF}} = \sum_{i,j} f(\overline{G}_\beta(i,j)) + \sum_i f(\overline{n}_{1\beta}(i)) + \sum_j f(\overline{n}_{2\beta}(j)),$$

where $\overline{n}_{1\beta}$ and $\overline{n}_{2\beta}$ are the mean-field indicators of the elements of S_1 and S_2 that are in correspondence and $\overline{G}_\beta$ forms the optimized transportation plan between S_1 and S_2, and $f(x) = -x\ln(x) - (1-x)\ln(1-x)$ for $x \in (0,1)$. The function f defines an entropy that is positive, bounded above by $\ln(2)$. Therefore, the entropy satisfies the following constraints:

$$0 \leqslant S_\beta^{\mathrm{MF}} \leqslant (N_1N_2 + N_1 + N_2)\ln(2). \tag{G1}$$

Using Eq. (17), after rearrangement we obtain,

$$U_\beta^{\mathrm{MF}} - \frac{1}{\beta}(N_1N_2 + N_1 + N_2)\ln(2) \leqslant F_\beta^{\mathrm{MF}} \leqslant U_\beta^{\mathrm{MF}}.$$

Taking the limits when $\beta \to +\infty$, we get

$$F^{\mathrm{MF}}(\infty) = U^{\mathrm{MF}}(\infty). \tag{G2}$$

■

2. Proof that $U^* \leqslant F^{\mathrm{MF}}(\infty)$

Let $U^{\mathrm{MF}}(\beta)$ be the mean-field internal energy at the inverse temperature β:

$$U_\beta^{\mathrm{MF}} = \sum_{i,j} C(i,j) X_\beta^{\mathrm{MF}}(i,j),$$

where X_β^{MF} is the solution the the SPA system of equations. At a finite inverse temperature β, X_β^{MF} is strictly non integral, as each of its terms is of the form $h(\beta(y(i,j)))$ where $h(x) = 1/(1+e^x)$, and therefore strictly in (0,1). In addition, X_β^{MF} satisfies constraints on row sums and column sums that makes it a doubly substochastic matrix with fixed total sum k. Based on Theorem 9, X_β^{MF} can be written as a linear combination of the partial permutation matrices of rank k,

$$X_\beta^{\mathrm{MF}} = \sum_{\pi\in P_{N_1,N_2}(k)} a_\pi \pi,$$

with all $a_\pi \in [0,1]$ and $\sum_{\pi\in P_{N_1,N_2}(k)} a_\pi = 1$. Therefore,

$$U_\beta^{\mathrm{MF}} = \sum_{i,j} C(i,j) X_\beta^{\mathrm{MF}}(i,j) = \sum_{\pi\in P_{N_1,N_2}(k)} a_\pi \sum_{i|f_\pi(i)\neq 0} C(i, f_\pi(i)), \tag{G3}$$

where f_π is the injection associated with π, see Eq. (A1). As U^* is the minimum matching cost over all possible partial permutations in $P_{N_1,N_2}(k)$, we have

$$\sum_{i|f_\pi(i)\neq 0} C(i, f_\pi(i)) \geqslant U^*. \tag{G4}$$

Combining Eqs. (G3) and (G4), we get

$$U_\beta^{\mathrm{MF}} \geqslant \sum_{\pi\in P_{N_1,N_2}(M)} a_\pi U^* \geqslant \left(\sum_{\pi\in P_{N_1,N_2}(M)} a_\pi\right) U^*,$$

from which we conclude that at each β,

$$U^* \leqslant U_\beta^{\mathrm{MF}}.$$

Therefore, $U^* \leqslant U^{\mathrm{MF}}(\infty)$ and, consequently, $U^* \leqslant F^{\mathrm{MF}}(\infty)$, based on Eq. (G2).

3. Proof that $U^* \geqslant F^{\mathrm{MF}}(\infty)$

Let us first recall the definition of the free energy,

$$F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x) = -\frac{1}{\beta}\left(\sum_i \ln\left(1 + e^{\beta\lambda(i)}\right) + \sum_j \ln\left(1 + e^{\beta\mu(j)}\right)\right) - \frac{1}{\beta}\sum_{i,j} \ln\left(1 + e^{-\beta y(i,j)}\right) - kx.$$

Note this property of limits:

$$\lim_{\beta\to+\infty} \frac{\ln(1 + e^{-a\beta})}{\beta} = \begin{cases} 0 & \text{if } a \geqslant 0, \\ -a & \text{if } a \leqslant 0. \end{cases}$$

Therefore,

$$\lim_{\beta\to+\infty} F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x) = -\sum_{i|\lambda(i)\geqslant 0} \lambda(i) - \sum_{j|\mu(j)\geqslant 0} \mu(j) + \sum_{(i,j)|y(i,j)\leqslant 0} y(i,j) - kx. \tag{G5}$$

Let us consider a permutation π in $P_{N_1,N_2}(k)$ and f_π its associated injection [see Eq. (A1)]. We can write

$$\sum_{i|f_\pi(i)\neq 0} C(i, f_\pi(i)) = \sum_{i|f_\pi(i)\neq 0} (C(i, f_\pi(i)) + \lambda(i) + \mu(f_\pi(i)) + x) - \sum_{i|f_\pi(i)\neq 0} \lambda(i) - \sum_{j\in \mathrm{Im}(f_\pi)} \mu(j) - kx,$$

where $\mathrm{Im}(f_\pi) = \{l \in [1, N_2] \mid \exists k \in [1, N_1],\, f_\pi(k) = l\}$. Note that $|\mathrm{Im}(f_\pi)| = k$. The equation above can be rewritten as

$$\sum_{i|f_\pi(i)\neq 0} C(i, f_\pi(i)) = \sum_{i|f_\pi(i)\neq 0} y(i, f_\pi(i)) - \sum_{i|f_\pi(i)\neq 0} \lambda(i) - \sum_{j\in \mathrm{Im}(f_\pi)} \mu(j) - kx.$$

For each index i, the summand included in the first term on the right is always larger or equal to the sum of all the corresponding terms that are negative:

$$\sum_{i|f_\pi(i)\neq 0} y(i, f_\pi(i)) \geqslant \sum_{i|y(i,j)\leqslant 0} y(i,j).$$

Similarly,

$$\sum_{i|f_\pi(i)\neq 0} \lambda(i) \leqslant \sum_{i|\lambda(i)\geqslant 0} \lambda(i), \qquad \sum_{j\in \mathrm{Im}(f_\pi)} \mu(j) \leqslant \sum_{j|\mu(j)\geqslant 0} \mu(j).$$

Therefore,

$$\sum_{i|f_\pi(i)\neq 0} C(i, f_\pi(i)) \geqslant \sum_{(i,j)|y(i,j)\leqslant 0} y(i,j) - \sum_{i|\lambda(i)\geqslant 0} \lambda(i) - \sum_{j|\mu(j)\geqslant 0} \mu(j) - kx,$$

i.e., using Eq. (G5),

$$\sum_{i|f_\pi(i)\neq 0} C(i, f_\pi(i)) \geqslant \lim_{\beta\to+\infty} F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x). \tag{G6}$$

Equation (G6) is valid for all partial permutations π in $P_{N_1,N_2}(k)$. It is therefore valid for the optimal permutation π^* that solves the unbalanced k-cardinality assignment problem. Since $U^* = \sum_{i|f_\pi(i)\neq 0} C(i, f_{\pi^*}(i))$, we have

$$U^* \geqslant \lim_{\beta\to+\infty} F_\beta(\boldsymbol{\lambda}, \boldsymbol{\mu}, x).$$

As this equation is true for all $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$, and x it is true in particular for $\boldsymbol{\lambda} = \boldsymbol{\lambda}^{\mathrm{MF}}$, $\boldsymbol{\mu} = \boldsymbol{\mu}^{\mathrm{MF}}$, and $x = x^{\mathrm{MF}}$, leading to

$$U^* \geqslant \lim_{\beta\to+\infty} F_\beta^{\mathrm{MF}} = F^{\mathrm{MF}}(\infty).$$

We have shown that $U^* \leqslant F^{\mathrm{MF}}(\infty)$ and $F^{\mathrm{MF}}(\infty) \leqslant U^*$, therefore $U^* = F^{\mathrm{MF}}(\infty)$. The corresponding result for the internal energy, $U^* = U^{\mathrm{MF}}(\infty)$ follows directly from Eq. (G2).

APPENDIX H: PROOF OF THEOREM 4: BOUNDS ON THE ENTROPY, INTERNAL ENERGY, AND FREE ENERGY

1. Bounds on the entropy

In Appendix G, we have shown that [Eq. (G1)]:

$$0 \leqslant S_\beta^{\mathrm{MF}} \leqslant (N_1N_2 + N_1 + N_2)\ln(2).$$

We define $A(N_1, N_2) = (N_1N_2 + N_1 + N_2)\ln(2)$. While this is not a tight bound, it will be enough for all subsequent properties.

2. Bounds on the free energy

In Appendix E, we have shown that [see Eq. (E4)]

$$\beta \frac{dF_\beta^{\rm MF}}{d\beta} = -F_\beta^{\rm MF} + U_\beta^{\rm MF}.$$

Using this equation and the relationship between free energy, energy, and entropy at SPA [see Eq. (17)], we obtain

$$\frac{dF_\beta^{\rm MF}}{d\beta} = \frac{1}{\beta^2} S_\beta^{\rm MF}.$$

From the bounds on the entropy,

$$0 \leqslant \frac{dF_\beta^{\rm MF}}{d\beta} \leqslant \frac{A(N_1, N_2)}{\beta^2}.$$

By integrating over β between $+\infty$ and β,

$$0 \leqslant F^{\rm MF}(\infty) - F_\beta^{\rm MF} \leqslant \frac{A(N_1, N_2)}{\beta}.$$

Finally, as $F^{\rm MF}(\infty) = U^*$,

$$U^* - \frac{A(N_1, N_2)}{\beta} \leqslant F_\beta^{\rm MF} \leqslant U^*. \tag{H1}$$

3. Bounds on the energy

As $U_\beta^{\rm MF} = F_\beta^{\rm MF} + \frac{1}{\beta} S_\beta^{\rm MF}$, using the inequalities in Eqs. (G1) and (H1), we get

$$U_\beta^{\rm MF} \leqslant U^* + \frac{A(N_1, N_2)}{\beta}. \tag{H2}$$

In addition, $U_\beta^{\rm MF}$ is monotonic, decreasing, with limit U^* as $\beta \to +\infty$, $U^* \leqslant U_\beta^{\rm MF}$. Therefore,

$$U^* \leqslant U_\beta^{\rm MF} \leqslant U^* + \frac{A(N_1, N_2)}{\beta}.$$

APPENDIX I: PROOF OF THEOREM 5: BOUNDS ON THE k-ASSIGNMENT MATRIX $\overline{G}_\beta$

This proof is inspired by the proof of Theorem 6 in Appendix 2 of Ref. [45] and by Appendix F of Ref. [16].

We first recall that $\overline{G}_\beta$, is a doubly substochastic matrix with $\sigma(\overline{G}_\beta) = k$. As such, it can be written as a linear combination of the partial permutation matrices of rank k, $\pi \in P_{N_1,N_2}(k)$,

$$\overline{G}_\beta = \sum_{\pi \in P_{N_1,N_2}(k)} a_\pi \pi,$$

with all $a_\pi \in [0, 1]$ and $\sum_{\pi \in P_{N_1,N2}(k)} a_\pi = 1$.

We assume that the unbalanced k-cardinality assignment problem considered has a unique solution. We want to prove that $\max_{i,j} |\overline{G}_\beta(i, j) - G^*(i, j)| \leqslant \frac{A(N_1,N_2)}{\beta\Delta}$, where G^* is the optimal solution of the unbalanced k-cardinality assignment problem, $\Delta = U^{2*} - U^*$ the difference in total cost between the second best solution and the optimal solution, and $A(N_1, N_2) = (N_1N_2 + N_1 + N_2)\ln(2)$. We use for that a proof by contradiction. We assume that there exists a pair (i, j) such that

$$\frac{A(N_1, N_2)}{\beta\Delta} < |\bar{G}_\beta(i, j) - G^*(i, j)|.$$

Let us denote $B(i, j) = |\bar{G}_\beta(i, j) - G^*(i, j)|$. As G^* is a partial permutation matrix, $G^*(i, j) = 0$ or $G^*(i, j) = 1$.

In the first case,

$$B(i, j) = \bar{G}_\beta(i, j) = \sum_{\pi \in P_{N_1,N_2}(k)} a_\pi \pi(i, j).$$

Since G^* is a partial permutation matrix of rank k, it is included in the decomposition of $\overline{G}_\beta$, and therefore,

$$B(i, j) = a_{G^*} G^*(i, j) + \sum_{\pi \in P_{N_1,N_2}(M) - \{G^*\}} a_\pi \pi(i, j) = \sum_{\pi \in P_{N_1,N_2}(k) - \{G^*\}} a_\pi \pi(i, j) < \sum_{\pi \in P_{N_1,N_2}(k) - \{G^*\}} a_\pi = 1 - a_{G^*},$$

where the final equality follows from the fact that the sum of all coefficients a is equal to 1.

In the second case, $G^*(i,j)=1$,

$$B(i,j)=1-\overline{G}_\beta(i,j)=1-\sum_{\pi\in P_{N_1,N_2}(k)} a_\pi \pi(i,j).$$

Again, as G^* is included in the decomposition of $\overline{G}_\beta$,

$$B(i,j)=1-a_{G^*}G^*(i,j)-\sum_{\pi\in P_{N_1,N_2}(k)-\{G^*\}} a_\pi\pi(i,j)=1-a_{G^*}-\sum_{\pi\in P_{N_1,N_2}(k)-\{G^*\}} a_\pi\pi(i,j)<1-a_{G^*},$$

where the final inequality follows from the fact that $\sum_{\pi\in P_{N_1,N_2}(k)-\{G^*\}} a_\pi\pi(i,j)$ is positive.

In conclusion, in both cases, we have

$$\frac{A(N_1,A_2)}{\beta\Delta}<1-a_{G^*}. \tag{I1}$$

Now, let us look at the energy associated with $\bar{G}_\beta$:

$$U_\beta^{\mathrm{MF}}=\sum_{i,j}C(i,j)\overline{G}_\beta(i,j)=\sum_{\pi\in P_{N_1,N_2}(k)} a_\pi \sum_{i|f_\pi(i)\neq 0} C(i,f_\pi(i))=a_{G^*}U^*+\sum_{\pi\in P_{N_1,N_2}(k)-\{G^*\}} a_\pi \sum_{i|f_\pi(i)\neq 0} C(i,f_\pi(i))$$

$$\geqslant a_{G^*}U^*+\left(\sum_{\pi\in P_{N_1,N_2}(k)-\{G^*\}} a_\pi\right)U^{2*}\geqslant a_{G^*}U^*+(1-a_{G^*})U^{2*}\geqslant U^*+(1-a_{G^*})\Delta.$$

In Theorem 4, we have shown that

$$U^*\leqslant U_\beta^{\mathrm{MF}}\leqslant U^*+\frac{A(N_1,N_2)}{\beta}.$$

Therefore,

$$U^*+(1-a_{G^*})\Delta\leqslant U^*+\frac{A(N_1,N_2)}{\beta},$$

i.e.,

$$(1-a_{G^*})\leqslant\frac{A(N_1,N_2)}{\beta\Delta}, \tag{I2}$$

as Δ is strictly positive.

We have shown that $\frac{A(N_1,N_2)}{\beta\Delta}<1-a_{G^*}$ [Eq. (I1)] and $(1-a_{G^*})\leqslant\frac{A(N_1,N_2)}{\beta\Delta}$ [Eq. (I2)], i.e., we have reached a contradiction. Our hypothesis is wrong, and therefore $\max_{i,j}|\bar{G}_\beta(i,j)-G^*(i,j)|\leqslant\frac{A(N_1,N_2)}{\beta\Delta}$.

APPENDIX J: PROOF OF THEOREM 7: TERMINATION CRITERIA FOR THE GENERIC ASSIGNMENT PROBLEM

Let us start by proving the following lemma.

Lemma 1. Let S_1 and S_2 be two sets of points with cardinalities N_1 and N_2 and let C be a real-valued cost matrix between S_1 and S_2. Let G be a transportation matrix between S_1 and S_2 with the sums of all rows and columns smaller or equal to 1, and a total sum of k, namely G is a doubly substochastic matrix with fixed total sum. Let $U(G,C)$ be the total assignment cost associated with G, namely $U(G,C)=\sum_{k,l}C(k,l)G(k,l)$. Let $\mathbf{a}$ and $\mathbf{b}$ be any two real-valued vectors of size N_1 and N_2, respectively, and let c be a real number. Let $D_{\mathbf{a},\mathbf{b},c}$ be the matrix defined as $D_{\mathbf{a},\mathbf{b},c}(i,j)=C(i,j)+a(i)+b(j)+c$ for $(i,j)\in[1,N_1]\times[1,N_2]$. Then,

$$U(G,D_{\mathbf{a},\mathbf{b},c})=U(G,C)+m,$$

where $m=\sum_i a(i)+\sum_j b(j)+ck$ is a constant, independent of G.

Proof. Let $n_1(i)=\sum_{j=1}^{N_2}G(i,j)$ and $n_2(j)=\sum_{i=1}^{N_1}G(i,j)$, and let $D_1=\mathrm{diag}(n_1)$ and $D_2=\mathrm{diag}(n_2)$. We consider an augmented balanced assignment problem of size N_1+N_2. We define the matrix C^a as

$$C^a=\begin{bmatrix}C & 0\\ 0 & 0\end{bmatrix}$$

and G^a as [see Eq. (A2)]

$$G^a=\begin{bmatrix}G & I-D_1\\ I-D_2 & G^T\end{bmatrix}.$$

Both matrices C^a and G^a are square matrices of size $(N_1+N_2)\times(N_1+N_2)$. In addition, it can be shown that G^a satisfies

$$\sum_{l=1}^{N_1+N_2} G^a(k,l)=1, \qquad \sum_{k=1}^{N_1+N_2} G^a(k,l)=1,$$

i.e., G^a is a doubly stochastic matrices.

It is easy to see that

$$U(G,C)=U(G^a,C^a). \tag{J1}$$

Let us now define the augmented vectors $\mathbf{a}^a=(\mathbf{a},0,\ldots,0)$ and $\mathbf{b}^a=(\mathbf{b},0,\ldots,0)$ both of size N_1+N_2, and the matrix $D^a_{\mathbf{a},\mathbf{b},c}(k,l)=C^a(k,l)+a^a(k)+b^a(l)+c$, for $(k,l)\in[1,N_1+N_2]^2$. This matrix can be written as

$$D^a_{\mathbf{a},\mathbf{b},c}=\begin{bmatrix} D_{\mathbf{a},\mathbf{b},c} & cE_1 \\ cE_2 & cE_3^T \end{bmatrix},$$

where E_1 and E_2 are square matrices of size $N_1\times N_1$ and $N_2\times N_2$, respectively, and E_3 is a rectangular matrix of size $N_1\times N_2$. All elements of these three matrices are 1. We compute $U(G^a,D^a_{\mathbf{a},\mathbf{b},c})$ is two different ways.

(i) *Direct product of G^a with D^a.* From the definitions of G_a and $D^a_{\mathbf{a},\mathbf{b},c}$, we get

$$U\left(G^a,D^a_{\mathbf{a},\mathbf{b},c}\right)=U(G,D_{\mathbf{a},\mathbf{b},c})+c\sum_{i,i'}(I(i,i')-D_1(i,i'))E_1(i,i')+c\sum_{j,j'}(I(j,j')-D_2(j,j'))E_1(j,j')+c\sum_{i,j}G(i,j)E_3(i,j),$$

where the sums extend over all $(i,i')\in[1,N_1]^2$ and $(i,j')\in[1,N_2]^2$. Considering the values of the matrices and the fact that G is a doubly substochastic matrix of total sum k,

$$U\left(G^a,D^a_{\mathbf{a},\mathbf{b},c}\right)=U(G,D_{\mathbf{a},\mathbf{b},c})+c\left(\sum_i I(i,i)-\sum_i n_1(i)\right)+c\left(\sum_j I(j,j')-\sum_j n_2(j)\right)+c\sum_{i,j}G(i,j),$$

i.e.,

$$U\left(G^a,D^a_{\mathbf{a},\mathbf{b},c}\right)=U(G,D_{\mathbf{a},\mathbf{b},c})+c\left(N_1-\sum_{i,j}G(i,j)\right)+c\left(N_2-\sum_{i,j}G(i,j)\right)+c\sum_{i,j}G(i,j).$$

Considering the values of the matrices and the fact that G is a doubly substochastic matrix of total sum k, we get

$$U\left(G^a,D^a_{\mathbf{a},\mathbf{b},c}\right)=U(G,D_{\mathbf{a},\mathbf{b},c})+c(N_1+N_2)-ck. \tag{J2}$$

(i) *Using the definition of the matrix $D^a_{\mathbf{a},\mathbf{b},c}$:*

$$\begin{aligned} U\left(G^a,D^a_{\mathbf{a},\mathbf{b},c}\right)&=\sum_{i,j=1}^{N_1+N_2}(C^a(i,j)+a^a(i)+b^a(j)+c)G^a(i,j)\\ &=U(G^a,C^a)+\sum_{i=1}^{N_1+N_2}a^a(i)\sum_{j=1}^{N_1+N_2}G^a(i,j)+\sum_{j=1}^{N_1+N_2}b^a(j)\sum_{i=1}^{N_1+N_2}G^a(i,j)+\sum_{i,j=1}^{N_1+N_2}cG^a(i,j). \end{aligned}$$

G^a is a doubly stochastic matrix of size $(N_1+N_2)^2$. Therefore,

$$U\left(G^a,D^a_{\mathbf{a},\mathbf{b},c}\right)=U(G^a,C^a)+\sum_{i=1}^{N_1+N_2}a^a(i)+\sum_{j=1}^{N_1+N_2}b^a(j)+c(N_1+N_2)=U(G^a,C^a)+\sum_{i=1}^{N_1}a(i)+\sum_{l=1}^{N_2}b(j)+c(N_1+N_2). \tag{J3}$$

Combining Eqs. (J1)–(J3), we get

$$U(G,D_{\mathbf{a},\mathbf{b},c})=U(G,C)+\sum_{i=1}^{N_1}a(i)+\sum_{j=1}^{N_2}b(j)+ck, \tag{J4}$$

which concludes the proof. ■

It is clear from Lemma 1 that solving the unbalanced k-cardinality assignment problem between S_1 and S_2 with the cost matrix C is equivalent to solving the k-cardinality assignment problem with the cost matrix $D_{\mathbf{a},\mathbf{b},c}$. This is generally of no use, with one significant exception, the setting of Theorem 7.

Indeed, let us consider an inverse temperature β and let $\boldsymbol{\lambda}^{\mathrm{MF}}$, $\boldsymbol{\mu}^{\mathrm{MF}}$, and x^{MF} be the mean-field solutions of the k-cardinality assignment problem at that temperature. Let us suppose that the corresponding matrix $\tilde{G}^{\mathrm{MF}}_\beta$ contains exactly k values that are

greater than $\frac{1}{2}$. Since $\bar{G}_\beta^{\mathrm{MF}}$ is a solution to the k-cardinality assignment problem, $\sum_i \bar{G}_\beta^{\mathrm{MF}}(i,j) \leqslant 1$ and $\sum_j \bar{G}_\beta^{\mathrm{MF}}(i,j) \leqslant 1$. Therefore, no two of the values fall on the same row, and similarly no two of those values fall on the same column. We can then define a function π such that $\pi(i) = j$ if (i,j) are the two indices of one of the k values considered, and $\pi(i) = 0$ otherwise. π is in fact a partial permutation of rank k.

We know that $\bar{G}_\beta^{\mathrm{MF}}(i,j) = h(y^{\mathrm{MF}}(i,j))$, where $y^{\mathrm{MF}}(i,j) = C(i,j) + \lambda^{\mathrm{MF}}(i) + \mu^{\mathrm{MF}}(j) + x^{\mathrm{MF}}$. We also know that $0 < h(x) \leqslant \frac{1}{2}$ when $x \geqslant 0$ and $\frac{1}{2} \leqslant h(x) < 1$ when $x \leqslant 0$. Therefore,

$$\begin{aligned} y^{\mathrm{MF}}(i,\pi(i)) &< 0 \quad \text{when } \pi(i) \neq 0, \\ y^{\mathrm{MF}}(i,j) &> 0 \quad \text{otherwise.} \end{aligned}$$

By setting the vectors $\mathbf{a}$ and $\mathbf{b}$ and the constant c in Lemma 1 to be $\boldsymbol{\lambda}^{\mathrm{MF}}$, $\boldsymbol{\mu}^{\mathrm{MF}}$, and x^{MF}, respectively, we have $D_{\boldsymbol{\lambda}^{\mathrm{MF}},\boldsymbol{\mu}^{\mathrm{MF}},x^{\mathrm{MF}}}(i,j) = y^{\mathrm{MF}}(i,j)$, and, therefore,

$$\begin{aligned} D_{\boldsymbol{\lambda}^{\mathrm{MF}},\boldsymbol{\mu}^{\mathrm{MF}},x^{\mathrm{MF}}}(i,\pi(i)) &< 0 \quad \text{when } \pi(i) \neq 0, \\ D_{\boldsymbol{\lambda}^{\mathrm{MF}},\boldsymbol{\mu}^{\mathrm{MF}},x^{\mathrm{MF}}}(i,j) &> 0 \quad \text{otherwise.} \end{aligned}$$

The unbalanced assignment problem of size k associated with this cost matrix $D_{\boldsymbol{\lambda}^{\mathrm{MF}},\boldsymbol{\mu}^{\mathrm{MF}},x^{\mathrm{MF}}}$ is then simple: element i in S_1 is trivially associated with element $\pi(i)$ in S_2 when $\pi(i) \neq 0$, as the corresponding cost is negative and therefore minimal compared to all the other costs that are positive. Its solution is then the partial permutation matrix π, and based on Lemma 1, it is also the solution to the original unbalanced k-cardinality assignment problem. The matrix G^* can then be obtained by rounding off the elements of $\bar{G}_\beta^{\mathrm{MF}}$ to the nearest integer.

[1] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2009).

[2] M. Dell'Amico and S. Martello, Discrete Appl. Math. **76**, 103 (1997).

[3] L. Ramshaw and R. E. Tarjan, Tech. Rep. No. HPL-2012-40R1 (HP Labs, Palo Alto, CA, 2012).

[4] A. Volgenant, Eur. J. Oper. Res. **157**, 322 (2004).

[5] A. Rosenmann, J. Comb. Optim. **44**, 1265 (2022).

[6] K. Date and R. Nagi, Parallel Comput. **57**, 52 (2016).

[7] S. S. Yadav, P. A. C. Lopes, A. Ilic, and S. K. Patra, Int. J. Commun. Syst. **32**, e3884 (2019).

[8] P. Koehl, M. Delarue, and H. Orland, Phys. Rev. E **103**, 012113 (2021).

[9] G. Birkhoff, Univ. Nac. Tucuman, Ser. A **5**, 147 (1946).

[10] J. von Neumann, A certain zero-sum two-person game equivalent to the optimal assignment problem, in *Contributions to the Theory of Games (AM-28)*, Vol. II, edited by H. W. Kuhn and A. W. Tucker (Princeton University Press, Princeton, 1953), pp. 5–12.

[11] C. Villani, *Topics in Optimal Transportation*, Graduate Studies in Mathematics (American Mathematical Society, Providence, RI, 2003).

[12] C. Villani, *Optimal Transport: Old and New*, Grundlehren der mathematischen Wissenschaften (Springer, Berlin, Heidelberg, 2008).

[13] G. Peyré and M. Cuturi, arXiv:1803.00567.

[14] P. Koehl, M. Delarue, and H. Orland, Phys. Rev. Lett. **123**, 040603 (2019).

[15] P. Koehl, M. Delarue, and H. Orland, Phys. Rev. E **100**, 013310 (2019).

[16] P. Koehl and H. Orland, Phys. Rev. E **103**, 042101 (2021).

[17] C. C. Paige and M. A. Saunders, SIAM J. Numer. Anal. **12**, 617 (1975).

[18] NVIDIA, P. Vingelmann, and F. H. Fitzek, Cuda, release: 10.2.89 (2020).

[19] R. Chandrasekaran, S. Kabadi, and K. Murty, Oper. Res. Lett. **1**, 101 (1982).

[20] H. J. Greenberg, Naval Res. Logist. Quart. **33**, 635 (1986).

[21] N. Megiddo and R. Chandrasekaran, Oper. Res. Lett. **8**, 305 (1989).

[22] R. Silver, Commun. ACM **3**, 605 (1960).

[23] R. E. Machol and M. Wien, Oper. Res. **24**, 190 (1976).

[24] P. A. Krokhmal and P. M. Pardalos, Eur. J. Oper. Res. **194**, 1 (2009).

[25] D. Coppersmith and G. B. Sorkin, Random Struct. Algor. **15**, 113 (1999).

[26] S. E. Alm and G. B. Sorkin, Comb. Probab. Comput. **11**, 217 (2002).

[27] S. Linusson and J. Wästlund, Probab. Theory Relat. Fields **128**, 419 (2004).

[28] C. Nair, B. Prabhakar, and M. Sharma, Random Struct. Algor. **27**, 413 (2005).

[29] J. Wästlund, *A Simple Proof of the Parisi and Coppersmith-Sorkin Formulas for the Random Assignment Problem* (Linköping University Electronic Press, Linköping, Sweden, 2005).

[30] G. Parisi, arXiv:cond-mat/9801176.

[31] R. Parviainen, Comb. Probab. Comput. **13**, 103 (2004).

[32] R. E. Machol and M. Wien, Oper. Res. **24**, 364 (1977).

[33] M. Dell'Amico and P. Toth, Discrete Appl. Math. **100**, 17 (2000).

[34] J. Munkres, J. Soc. Ind. Appl. Math. **5**, 32 (1957).

[35] R. Pilgrim, Tutorial on implementation of Munkres' assignment algorithm (1995), doi: 10.13140/RG.2.1.3572.3287.

[36] https://github.com/faolane/LAP.

[37] R. Jonker and T. Volgenant, Computing **38**, 325 (1987).

[38] https://github.com/yongyanghz/LAPJV-algorithm-c/.

[39] M. Dell'Amico, A. Lodi, and S. Martello, Discrete Appl. Math. **110**, 25 (2001).
[40] https://site.unibo.it/operations-research/en/research/library-of-codes-and-instances-1.
[41] P. A. Lopes, S. S. Yadav, A. Ilic, and S. K. Patra, J. Parallel Distrib. Comput. **130**, 50 (2019).
[42] https://github.com/paclopes/HungarianGPU.
[43] R. D. Team, *RAPIDS: Libraries for End to End GPU Data Science* (2023), doi:10.2196/preprints.23246.
[44] https://github.com/rapidsai.
[45] J. Kosowsky and A. Yuille, Neural Netw. **7**, 477 (1994).
[46] L. Mirsky, Archiv der Mathematik **10**, 88 (1959).
[47] P. Čihák, Commentationes Mathematicae Universitatis Carolinae **11**, 385 (1970).
[48] N. S. Mendelsohn and A. L. Dulmage, Proc. Am. Math. Soc. **9**, 253 (1958).
[49] R. A. Brualdi and G. M. Lee, Linear Alg. Appl. **19**, 33 (1978).