Efficient algorithms to solve atom reconfiguration problems. I. The redistribution-reconfiguration (red-rec) algorithm

Barry Cimring,[1] Remy El Sabeh,[2] Marc Bacvanski,[3] Stephanie Maaz,[4] Izzat El Hajj,[2] Naomi Nishimura,[4] Amer E. Mouawad,[2,4,5] and Alexandre Cooper[1,*]

[1] *Institute for Quantum Computing, University of Waterloo, Waterloo, Ontario N2L 6R2, Canada.*
[2] *Department of Computer Science, American University of Beirut, Lebanon.*
[3] *Khoury College of Computer Science, Northeastern University, United States.*
[4] *David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Ontario N2L 6R2, Canada.*
[5] *University of Bremen, Bremen, Germany*

(Dated: October 27, 2022)

We propose the redistribution-reconfiguration (red-rec) algorithm to efficiently compute control protocols to assemble compact-centered configurations of atoms in two-dimensional arrays of optical traps with lattice geometries. The red-rec algorithm redistributes atoms among pairs of donor-receiver columns and reconfigures each column using an exact displacement-minimizing algorithm, harnessing parallel control operations that simultaneously actuate multiple traps to reduce the execution time. We numerically quantify the performance of the red-rec algorithm, both in the absence and in the presence of loss, using realistic physical parameters and operational constraints. We show that the number of traps required to prepare a compact-centered configuration of atoms on a grid with a mean success probability of one half scales as the 3/2 power of the number of desired atoms, highlighting the challenges of assembling configurations of tens of thousands of atoms. We further demonstrate that faster preparation times can be achieved by rejecting configurations of atoms containing fewer atoms than a given threshold. The red-rec algorithm admits an efficient implementation that can readily be deployed on real-time control systems to assemble large configurations of atoms with high mean success probability and fast preparation times.

I. INTRODUCTION

Quantum many-body systems formed by configurations of single quantum particles, such as neutral atoms, charged ions, and molecules, offer valuable features for quantum information processing [1–5]. These features include long-lived internal states to encode quantum information, tunable interaction mechanisms to create many-body quantum coherence, and experimentally-accessible optical and microwave transitions for trapping, cooling, and imaging single particles, as well as coherently manipulating their internal states. Operated as programmable quantum simulators [6–13], these systems realize lattice spin models that can be mapped onto various other physical models, providing access to their equilibrium and non-equilibrium properties in regimes otherwise inaccessible using classical simulation techniques.

A promising approach to prepare large controllable quantum many-body systems is trapping single quantum particles into arrays of optical traps generated from multiplexed laser beams tightly-focused through a high-resolution imaging system into a vacuum chamber [14–16]. Exploiting real-time feedback control systems together with active diffractive optical elements [17–20], configurations of hundreds of single atoms have been assembled by dynamically updating the geometry of the trap array [21–23] or displacing atoms from one trap to another using a secondary array of dynamic traps [24–31]. These feedback assembly techniques have solved the problem of deterministically preparing configurations of atoms in arrays of optical traps without relying on probabilistic sampling or a quantum phase transition [32–35]. Extending these techniques to displace atoms while preserving their internal many-body coherence has further enabled realizing quantum many-body systems with dynamic connectivity graphs, effectively lifting constraints imposed by spatial locality when implementing quantum information processing protocols [36, 37].

Computing sequences of control operations to prepare deterministic configurations of atoms from arbitrary ones requires solving atom reconfiguration problems [38, 39]. These problems are hard combinatorial optimization problems, not only because there exists an exponentially-large number of valid solutions, but also because they are *hidden stochastic* problems; atoms might be lost during control operations and the presence of an atom in each trap of the array is a random variable (*stochastic*) that remains unknown until a measurement is performed (*hidden*). These problems differ from *deterministic* reconfiguration problems [40–43], which have been studied in both fundamental, graph-theoretic contexts [44, 45], as well as operational, application-specific contexts [46, 47] for applications in robot motion planning [41, 47–49], network frequency assignment [50], and artificial intelligence [51].

Efficient exact and approximation algorithms for solving atom reconfiguration problems are known only for the simplest cases [39, 52], such as minimizing the total dis-

* alexandre.cooper@uwaterloo.ca

tance traveled by all atoms in the absence of loss using reconfiguration algorithms that solve assignment problems, e.g., using the Hungarian algorithm [53–55]. Experimental implementations have thus relied on exploiting heuristic algorithms [29, 38, 56–58] that trade off guarantees of optimal performance for operational simplicity and computational speed. These heuristic algorithms exploit *ad hoc* principles derived from intuition and experience to define specific control subroutines; such algorithms are particularly useful when the complexity of the problem is such that there are no known efficient algorithms to solve it. Preparing configurations of thousands of atoms with high success probability at a fast rate calls for the continuous development of heuristic algorithms with improved performance, e.g., by exploiting opportunities offered by parallel control operations. A formal framework to benchmark the performance of such heuristics against exact and approximation algorithms would also facilitate their shared development towards increasing the operational performance of programmable quantum simulators.

In this paper, we introduce and benchmark the redistribution-reconfiguration (red-rec) algorithm to prepare two-dimensional (2D) configurations of atoms with lattice geometries. The red-rec algorithm exploits simple heuristics, redistributing surplus atoms among columns before reconfiguring each column using an exact one-dimensional (1D) reconfiguration algorithm. The execution time of reconfiguration protocols is reduced by performing a sequence of control operations in parallel, whereas the mean wait time between the preparation of two successful configurations is reduced by rejecting initial configurations that contain fewer atoms than a given threshold.

The presentation of our results proceeds as follows. After introducing atom reconfiguration problems and the red-rec algorithm (Secs. II–III), we numerically quantify the performance of the red-rec algorithm in the absence of loss in terms of the number of control operations, and compare these values with those obtained using exact and approximation algorithms (Secs. IV–V). Our results indicate that the red-rec algorithm performs a number of displacement operations approaching the optimal value achieved by an assignment algorithm, albeit with some atoms being displaced more than once. We then quantify the performance of the red-rec algorithm in the presence of loss using realistic physical parameters and operational constraints (Sec. VI), focusing on restrictions imposed by the field of view, the total number of optical traps, and the loading efficiency. We finally introduce and characterize a thresholding strategy to increase duty cycle (Sec. VII) and conclude by reviewing key results and outlining opportunities to further improve the performance of the red-rec algorithm (Sec. VIII).

II. ATOM RECONFIGURATION PROBLEMS

We refer to an *atom* at $\vec{x} \in \mathbb{R}^3$ as $a(\vec{x}, \tilde{p})$, where $\tilde{p} \in [0,1]$ is the probability of detecting the atom by performing a perfect measurement at $\vec{x}$. A *perfect measurement* at $\vec{x} \in \mathbb{R}^3$, $\pi(\vec{x})$, detects an atom at $\vec{x}$ if and only if there is an atom at $\vec{x}$. If a perfect measurement detects an atom at $\vec{x}$, then the probability $\tilde{p}$ of $a(\vec{x}, \tilde{p})$ is updated from a probabilistic value $\tilde{p} \in [0,1]$ to a deterministic value $p \in \{0,1\}$. An atom $a(\vec{x}, p)$ with $p = 0$ is said to have been lost and can be disregarded.

A *configuration of atoms* $\tilde{\mathcal{C}} = \{a_j(\vec{x}_j, \tilde{p}_j)\}_{j=1}^{N_a}$ is defined as a collection of N_a atoms that would each be detected by a collection of perfect measurements performed at the location of each atom, $\Pi = \{\pi_j(\vec{x}_j)\}_{j=1}^{N_a}$. A *deterministic* configuration of atoms $\mathcal{C}$ is a collection of atoms whose detection probability is unity, i.e., $p_j = 1$ for all $1 \leq j \leq N_a$.

We denote a *trap* at $\vec{x} \in \mathbb{R}^3$ as $t(\vec{x}, \tilde{p})$, where $\tilde{p} \in [0,1]$ is the probability that the trap t contains an atom $a(\vec{x}, \tilde{p})$, which we set equal to the probability of detecting $a(\vec{x}, \tilde{p})$ at $\vec{x}$. Each trap is implicitly specified by its physical properties, which include its effective trap depth. We define a *trap array* $\mathcal{A}(V, S) = \{t(\vec{x}_j, \tilde{p}_j)\}_{j=1}^{N_t}$ as a collection of $N_t = |V|$ traps, where $V = \{\vec{x}_j\}_{j=1}^{N_t}$ specifies the *geometry* of the trap array and $S = \{\tilde{p}_j\}_{j=1}^{N_t}$ specifies the *occupation state* of the trap array. The trap array $\mathcal{A}$ contains a configuration of atoms $\tilde{\mathcal{C}} = \{a(x_j, \tilde{p}_j) \mid x_j \in V(\mathcal{A}),\ \tilde{p}_j > 0\}_{j=1}^{N_t}$ that can be mapped onto a deterministic configuration of atoms $\mathcal{C}$ by performing a perfect measurement at the location of each trap of the trap array. We distinguish a *dynamic trap array* from a *static trap array* depending on whether the location and physical properties of the traps can change with time. The set of all dynamic traps is chosen to contain the static trap array so that an atom in a static trap can be displaced to any other static trap by controlling the dynamic trap array.

In the remainder of this paper we focus our attention on trap arrays defined in the restricted plane $[0, L]^2 \subset \mathbb{R}^2$, the *field of view*, which effectively replaces $\mathbb{R}^3$ in the previous definitions. Typical geometries in $\mathbb{R}^2$ include (a) *Bravais lattices* (oblique, rectangular, centered rectangular, square, and hexagonal), which are specified by their origin and generator vectors, (b) *sub-lattices*, which are specified by their origin, generator vectors of the parent lattice, and coordinate numbers (indicating which elements in the parent lattice are present), and (c) *arbitrary geometries*, which are specified by a list of spatial coordinates.

More specifically, we focus our attention on trap arrays with a grid geometry, effectively solving the problem of preparing a target configuration of $N_a^T = \sqrt{N_a^T} \times \sqrt{N_a^T}$ atoms forming a square-compact region in the center of a static trap array with a square-lattice geometry and rectangular dimensions containing $N_t = N_t^x \times N_t^y$ traps. We choose the number of traps to be $N_t^x = \sqrt{N_a^T}$ and

$N_t^y = \eta\sqrt{N_a^T}$, so that $N_t = \eta N_a^T$, where $\eta = N_t/N_a^T$ is the *overhead factor* quantifying the excess in optical traps needed to compensate for atom loss and imperfect loading efficiency. We refer to the *internal region* or *target region* of the trap array as the set of static traps containing the target configuration of atoms and the *external region* as the set of all remaining static traps. The red-rec algorithm could be further extended to more general oblique lattices with rhomboidal boundaries, including triangular lattices with two generator vectors of equal norm oriented at a relative angle of $\pi/3$.

A. Reconfiguration protocols

An atom reconfiguration problem seeks a sequence of control operations to prepare a predetermined target configuration of atoms from an arbitrary configuration of atoms. An atom reconfiguration problem is said to be *solvable* if the target configuration C_T is reachable in a finite number of control steps, and *efficiently solvable* if it is solvable and the number of control steps in a solution is polynomial in the size of the static trap array. An *optimal solution* with respect to a given cost function is a solution that minimizes the cost function, such as the total number of displacement operations.

We refer to the solution of an atom reconfiguration problem as a *control protocol* $\mathcal{P} = (\Pi_0, \boldsymbol{T}_1, \Pi_1, \ldots, \boldsymbol{T}_K, \Pi_K)$, which is defined as an initial measurement Π_0 followed by sequence of K reconfiguration cycles, each cycle comprising a *measurement step*, Π_k, and an *actuation step*, $\boldsymbol{T}_k$. The kth *actuation step* executes a sequence of elementary control operations, $\boldsymbol{T}_k = (T_{k,1}, T_{k,2}, \ldots, T_{k,L_k})$, where $T_{k,l}$ is the lth elementary control operation of the kth reconfiguration cycle and L_k is the number of elementary control operations required to execute the kth control sequence. The kth *measurement step* reveals the configuration of atoms contained in the trap array, effectively projecting into a deterministic state $\mathcal{C}_k$ of N_a^k atoms. If the number of detected atoms is less than the number of desired atoms, $N_a^k < N_a^T$, then the reconfiguration problem is said to be unsolvable, the protocol is said to have failed, and the execution of the protocol is aborted. On the other hand, if the number of detected atoms is greater than or equal to the number of desired atoms, $N_a^k \geq N_a^T$, then the reconfiguration problem is said to be solvable and the execution of the protocol continues. If the measured configuration contains the target configuration, $\mathcal{C}_k \supseteq \mathcal{C}_T$, then the problem is said to have been solved in $K = k$ reconfiguration cycles, the protocol is said to have succeeded, and the execution of the protocol is terminated. The reconfiguration protocol is thus executed until the target configuration has been reached or is no longer reachable because the number of remaining atoms is no longer sufficient. The number of reconfiguration cycles required to successfully prepare the target configuration of atoms thus depends on the initial configuration of atoms, as well as the realization of loss processes associated with control errors.

Each elementary control operation $T_{k,l}$ is chosen from a set of six elementary control operations, $\{T_{\alpha,\nu}^{\pm,0}\}$, defined as *transfer operations*, $T_\alpha^\pm$, which include *extraction operations*, T_α^+, and *implantation operations*, T_α^-, that transfer atoms from and to the static trap array to and from the dynamic trap array; *displacement operations*, $T_{\nu_j}^\pm$, that displace a dynamic trap by an elementary step along the generator vector $\vec{k}_j$ of the static trap array, either forward, $T_{\nu_j}^+$, or backward, $T_{\nu_j}^-$; and *no-op operations*, $\{T_\alpha^0, T_\nu^0\}$, that leave some traps undisturbed while transfer and displacement operations are performed on other traps. Explicitly defining no-op operations facilitates accounting for the loss of idle atoms in both static and dynamic traps. Given that they satisfy hardware constraints, the control operations acting on different individual traps can be executed in parallel by batching them into a sequence of control operations acting on multiple traps simultaneously.

The configuration of atoms contained in the static trap array is typically updated by extracting it from a static trap into a dynamic trap (extraction), displacing it by elementary steps to its target location (displacement), and implanting it back into a static trap (implantation); we refer to this sequence of operations as an extraction-displacement-implantation (EDI) cycle. Multiple dynamic traps can be actuated simultaneously, given that they satisfy specific operational constraints. In the present study, we restrict our analysis to dynamic trap arrays that form horizontal or vertical chains along rows or columns of a grid.

B. Control operations

Control operations might update both the state and the geometry of the static trap array, $A_s(V_s, S_s)$, and the dynamic trap array, $A_d(V_d, S_d)$. Transfer operations $T_\alpha^\pm(\vec{x}_l)$ update the state of the traps at $\vec{x}_l$ in both the static and dynamic trap arrays. Given a static trap $t_s(\vec{x}, \tilde{p})$ at $\vec{x} = \vec{x}_l$ containing an atom with probability $\tilde{p}$ and an overlaid dynamic trap $t_d(\vec{x}, 0)$ at $\vec{x}$ containing no atom, an extraction operation transfers an atom from the static trap to a dynamic trap, updating the static and dynamic traps from $t_s(\vec{x}, \tilde{p})$ and $t_d(\vec{x}, 0)$ to $t_s(\vec{x}, 0)$ and $t_d(\vec{x}, \tilde{p})$, respectively. Similarly, an implantation operation transfers an atom from a dynamic trap to a static trap, updating the static and dynamic traps from $t_s(\vec{x}, 0)$ and $t_d(\vec{x}, \tilde{p})$ to $t_s(\vec{x}, \tilde{p})$ and $t_d(\vec{x}, 0)$, respectively. A transfer operation acting on a pair of empty traps returns empty traps (no atom can be created spontaneously), whereas a transfer operation acting on a pair of occupied traps returns a pair of empty traps. These definitions assume that the static and dynamic traps are overlaid; otherwise, implanting an atom in free space leads to its loss. Similarly, elementary displacement operations $T_{\nu_j}^\pm$

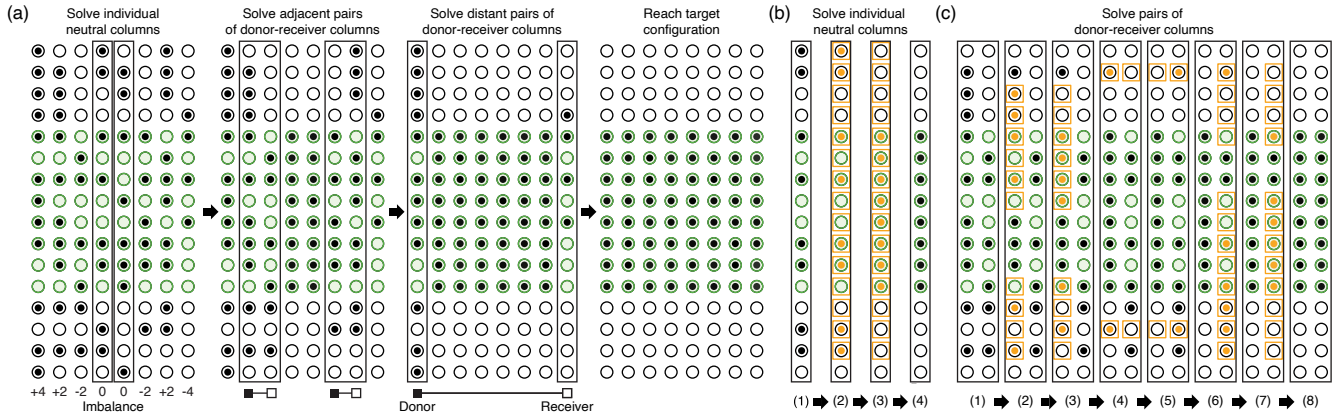

FIG. 1. **Single realization of the red-rec algorithm in the absence of loss.** (a) Sequence of control operations to prepare a target configuration of $N_a^T = 8 \times 8$ atoms (green shaded disks) in a static trap array of $N_t = 8 \times 16 = 128$ traps (black circles) from an arbitrary configuration of $N_a^0 = 64$ atoms (black dots). After computing the imbalance of each column as the difference between the number of detected and desired atoms, the algorithm sequentially reconfigures individual neutral columns, neighboring pairs of donor-receiver columns, and then distant pairs of donor-receiver columns. (b) An individual neutral column is solved by extracting atoms from the static trap array into the dynamic trap array (orange squares), displacing the atoms to the desired target locations, and implanting the atoms back into the static trap array. (c) A pair of donor-receiver columns is solved using a streamlined sequence of control operations that ((1)-(3)) reconfigures the donor column while displacing atoms to be exchanged to open rows, ((4)-(5)) redistributes surplus atoms from the donor column to the receiver column in a single parallel displacement step, and ((6)-(8)) reconfigures the receiver column.

acting on the dynamic trap $t_d(\vec{x}, \tilde{p})$ at $\vec{x}$ will displace the trap and the atom it contains from $\vec{x}$ to $\vec{x} \pm \vec{k}_j$, where k_j is the generator vector of the static trap array, e.g., $\vec{k}_j \in \{\vec{k}_x, \vec{k}_y\}$ on the grid.

Each elementary control operation acting on a trap might induce the loss of the atom it contains. We define the survival probability as p_α, p_ν, and p_0 for transfer, displacement, and no-op operations, respectively, assuming the no-op loss probability to be the same for both static and dynamic traps. We assume that control errors are limited to loss, that is, we ignore the probability of an atom being left behind in its original trap during a transfer operation. We refer to the *corruption* of an atom as the cumulative probability of it being lost given its control history; the corruption thus corresponds to the probability of detecting the atom using a perfect measurement.

To simplify experimental operations, prevent collisions between atoms, and ensure compatibility with typical actuating optical devices like acousto-optic deflectors, we restrict parallel control operations to linear chains of traps, corresponding to horizontal sub-rows or vertical sub-columns on the grid of the static trap array. We further restrict displacement operations to discrete elementary-step translations executed in constant time along the generator vectors of the static trap array. Greater performance might be achieved by executing parallel control operations on contiguous or non-contiguous 2D blocks of atoms, as well as displacing atoms along arbitrary trajectories at variable speed, albeit at the expense of more complex heuristics.

III. THE RED-REC ALGORITHM

The red-rec algorithm solves atom reconfiguration problems on grids by breaking them down into a sequence of simpler reconfiguration problems on individual columns and pairs of columns (Fig. 1a).

First, following each measurement, the algorithm computes the imbalance of atoms within each column of the trap array as the difference between the number of measured atoms and the number of desired atoms. Columns are thus classified as donor (positive imbalance), receiver (negative imbalance), or neutral (zero imbalance).

Second, the algorithm reconfigures individual neutral columns using an exact reconfiguration algorithm on 1D chains (see App. B), which identifies the subset of atoms whose reconfiguration minimizes the total number of displacement operations in a single EDI cycle (Fig. 1b). Because the column is neutral, the reconfiguration problem is trivially solved by assigning surplus atoms to empty traps from the outside to the inside of the trap array.

Third, the algorithm reconfigures pairs of donor-receiver columns starting with adjacent pairs that can fully satisfy one another, followed by adjacent and distant pairs that can exchange the most atoms in a single redistribution sequence. The pairs of donor-receiver columns are solved using a streamlined sequence of operations, which redistributes atoms from donor columns to receiver columns (Fig. 1c). The atoms within the donor column are labeled according to three categories: *reconfigured atoms*, which are to populate the target region of the donor column, *redistributed atoms*, which are to populate the target region of the receiver column, and

idle atoms, which are to remain idle within the donor column (we have numerically confirmed that placing idle atoms near the target region to facilitate replacing lost atoms in subsequent reconfiguration cycles is detrimental, resulting in a $1-2$ % reduction in the mean success probability). Reconfigured atoms are chosen among the atoms located in both external and target regions to populate the target region with the smallest number of displacement operations. Redistributed atoms are chosen exclusively among the atoms located in the external region, starting from the ones located the farthest away from the target region, favoring the choice of an equal numbers of atoms above and below the target region.

The redistributed atoms are assigned to the nearest open rows along which they can be displaced from the donor column to the receiver column without obstructions, prioritizing rows closer to the target region. This assignment is done greedily by going over each redistributed atom one by one and, for each one of them, choosing the closest open row or the one closer to the target region if there are two closest rows. The redistributed atoms transferred to the receiver column are reconfigured by solving the exact 1D reconfiguration algorithm. We restrict redistribution along rows in the external regions to avoid increasing the corruption of atoms in the centre of the target region. To decrease the total number of displacement operations, at the cost of increasing the computational time, the labeling of redistributed atoms and their assignment to distribution rows could be achieved by exhaustively searching over all possibilities.

The streamlined sequence of operations is used to reconfigure the donor column while bringing the redistributed atoms to open rows, distribute redistributed atoms from the donor to the receiver column in parallel, and reconfigure the receiver column (Fig. 1c). The streamlined sequence of operations only extracts and implants the redistributed atoms once, thus reducing the number of transfer operations. If not all redistributed atoms can be redistributed in a single step, e.g., because the number of open rows is less than the number of atoms to be redistributed, then the streamlined sequence of operations is repeated until the receiver column reaches zero imbalance. If the receiver column has not been fully satisfied by the donor column, then then it is paired with another donor column. If there are no open rows along which to distribute atoms between any of the donor and receiver columns, where an open row is a row that has no obstructing atoms on the trajectory between the donor column and the receiver column, the algorithm reconfigures all the individual columns and restarts the pairing.

Fourth, a measurement is performed to uncover the configuration of atoms in the static trap array. The measurement-reconfiguration cycle is repeated until the reconfiguration problem succeeds or fails, i.e., the measured configuration contains the target configuration, or too few atoms remain to achieve the target configuration.

IV. BENCHMARKING AGAINST EXACT AND APPROXIMATION ALGORITHMS

To obtain bounds on the minimum number of displacement and transfer operations required to solve a given atom reconfiguration problem, we benchmark the red-rec algorithm against the minimum-weight perfect matching algorithm (MWPM), which minimizes the total number of displacement operations, and the Steiner-tree 3-approximation algorithm (3-approx), which bounds the total number of transfer operations by at most three times its optimal value (see App. A for a description of these two algorithms).

In the absence of loss, we quantify the performance of a reconfiguration algorithm to prepare a given target configuration of atoms in terms of the mean number of displacement and transfer operations averaged over randomly-sampled initial configurations (1,000 realizations). We specifically focus on the problem of preparing a square-compact configuration of $N_a^T = \sqrt{N_a^T} \times \sqrt{N_a^T}$ atoms in the center of rectangular array of $N_t = \sqrt{N_a^T} \times 2\sqrt{N_a^T}$ traps, i.e., with twice as many traps as desired atoms ($\eta = 2$). We set the number of surplus atoms to zero by choosing the number of atoms in the initial configuration to be equal to the number of atoms in the target configuration, i.e., $N_a^0 = N_a^T$.

The red-rec algorithm performs slightly more displacement operations than the optimal value achieved by the MWPM algorithm; the relative numbers of displacement operations decrease with increasing system size and reach a value of 1.04(1) for a configuration of $N_a^T = 32^2 = 1024$ atoms (Fig. 2a). The slight excess in displacement operations results from a small fraction of atoms, such as those redistributed across distant pairs of donor-receiver columns, undergoing up to twice as many displacements as the maximum per-atom displacement value measured for MWPM (Fig. 2b). The number of displacement operations is positively correlated with the number of transfer operations (Fig. 2c). Compared to the 3-approx algorithm, the red-rec algorithm performs 1.5 to 2.0 times more transfer operations. We note that transfer operations include both extraction and implantation operations, so that the total number of transfer operations is even by construction, that is, all extracted atoms are implanted back in the static trap array. The relative numbers of transfer operations increase with increasing system size, up to a value of 1.8(1) for a configuration of $N_a^T = 32^2$ atoms (Fig. 2a). Whereas the 3-approx algorithm leaves nearly half of the atoms idle, the red-rec algorithm moves nearly all atoms at least once as it sequentially reconfigures each column at least once with fewer than 7% of the columns getting reconfigured more than twice (Fig. 2d).

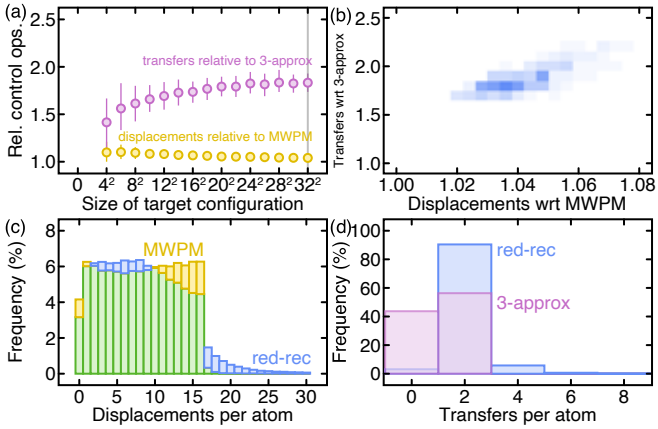

FIG. 2. **Relative numbers of control operations in the absence of loss.** (a) Relative numbers of transfer and displacement operations performed by the red-rec algorithm expressed relative to the 3-approx and MWPM algorithms, respectively, for varying configuration sizes at constant overhead factor $\eta = 2$ with no surplus atoms ($N_A^0 = N_A^T$). (b) Distribution of the number of transfer operations relative to 3-approx and displacement operations relative to MWPM for preparing a configuration of 32×32 atoms in an array of 32×64 traps. (c) Distribution of the number of displacement operations performed per atom for the red-rec (blue) and MWPM (yellow) algorithms. The MWPM algorithm exactly minimizes the number of displacement operations. (d) Distribution of the number of transfer operations per atom for the red-rec (blue) and 3-approx (purple) algorithms. The 3-approx algorithm achieves at most three times the minimum number of transfer operations.

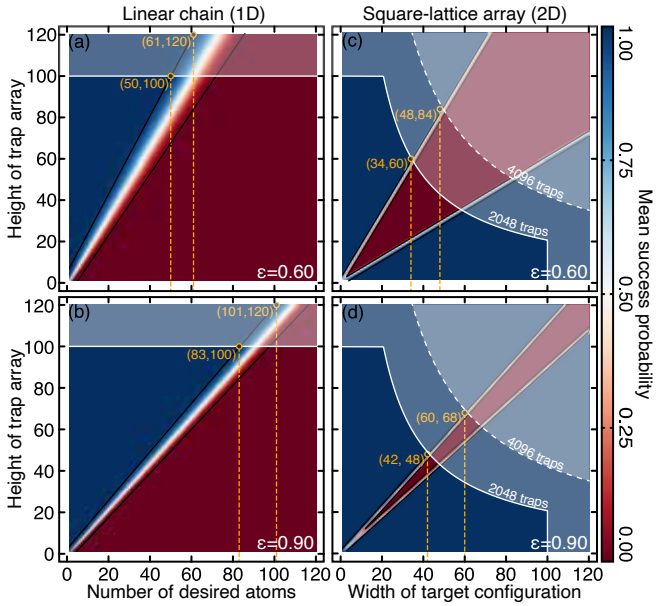

FIG. 3. **Baseline success probability in the absence of loss.** (a-b) Probability of successfully preparing a chain of N_a^T atoms in a linear array of N_t traps in the absence of loss for a loading efficiency of $\epsilon = 0.60$ and $\epsilon = 0.90$. The length of the chain is limited by the field of view to 100 traps (white region). The regions of near-certain failure ($\bar{p} \leq 0.02$, red) and near-certain success ($\bar{p} \geq 0.98$, blue) are delineated by transition curves for a number of traps that is linearly proportional to the number of desired atoms (black line). (c-d) Probability of successfully preparing a configuration of $N_a^T = \sqrt{N_a^T} \times \sqrt{N_a^T}$ atoms in a square-lattice array of $N_t^x \times N_t^y$ traps with $N_t^x = \sqrt{N_a^T}$ (width) in the absence of loss. The accessible parameter region is limited by the field of view (100 traps along each direction) and the total number of traps (white region).

V. BASELINE SUCCESS PROBABILITY IN THE ABSENCE OF LOSS

In the absence of loss, the probability of successfully preparing a given target configuration of atoms is fully determined by the number of atoms in the initial configuration, $N_a^0 \sim \text{Bino}(N_t, \epsilon)$, which follows a binomial distribution that is completely determined by the size of the static trap array, N_t, and the mean loading efficiency, ϵ. Typical values for the loading efficiency are $\epsilon = 0.60$ (0.50) for alkali (alkaline-earth) atoms, although $\epsilon \geq 0.60$ has been achieved using enhanced-loading techniques [59–61].

We define the *baseline success probability* as the probability that the initial configuration of atoms contains at least as many atoms as needed to prepare the target configuration of atoms, $p_0 = P(N_a^0 \geq N_a^T)$. The baseline success probability is thus given by the complementary cumulative distribution function of the binomial distribution, $p_0 = \text{BinoCCDF}(N_t, \epsilon; N_a^T)$, which has a known analytical expression. The baseline success probability provides an exact upper bound on the mean success probability in the presence of loss, as well as an estimate of the minimum number of static traps required to prepare a given target configuration of atoms with a given success probability.

We compute the baseline success probability of preparing linear-chain and square-lattice configurations of atoms for various sizes of static trap arrays, target configurations, and loading efficiencies (Fig. 3). The probability surfaces exhibit a region of near-certain success ($p_0 \geq 0.98$) separated from a region of near-certain failure ($p_0 \leq 0.02$) by a sharp transition region centered around the iso-probability line $p_0 = 0.5$ realized at the *baseline overhead factor* $\eta = N_t/N_a^T = 1/\epsilon$.

In linear chains of 100 (120) static traps, the largest centered-compact configuration of atoms that can be prepared with near-certain success contains $N_a^T = 50$ (61) atoms (Fig. 3a); increasing the loading efficiency to $\epsilon = 0.90$ increases this configuration size to 83 (101) atoms (Fig. 3b). In square-lattice trap arrays containing at most $N_t^x \times N_t^y = 2{,}048$ (4,096) optical traps, e.g., due to restrictions in optical power, diffraction and transmission efficiency, and polarizability, the largest centered-compact configuration of atoms that can be prepared with near-certain success contains $N_a^T = 34^2 = 1{,}156$ ($42^2 = 1{,}756$) atoms (Fig. 3c); increasing the loading efficiency to $\epsilon = 0.90$ increases the configuration size

to $N_a^T = 48^2 = 2{,}304$ ($60^2 = 3{,}600$) atoms (Fig. 3d).

An important bottleneck in scaling up atom configurations beyond a few thousand atoms is the limit on the number of optical traps. With a loading efficiency of $\epsilon = 0.60$, preparing a configuration of $100^2 = 10{,}000$ atoms with near-certain success would require at least $100 \times 330 = 33{,}000$ traps, not to mention the problem of atom loss that increases the scaling of the overhead factor from linear to quadratic (Fig. 5a). Another bottleneck is restrictions in the field of view. Assuming height limits of 100 (120) traps, the largest configuration of atoms that can be prepared with near-certain success contains $N_a^T = 58^2 = 3{,}364$ ($70^2 = 4{,}900$) atoms for $\epsilon = 0.60$ and $N_a^T = 88^2 = 7{,}744$ ($106^2 = 11{,}236$) atoms for $\epsilon = 0.90$.

VI. MEAN SUCCESS PROBABILITY IN THE PRESENCE OF LOSS

In the presence of loss, we quantify the performance of a reconfiguration algorithm in terms of the mean success probability, $\bar{p} \leq p_0$, which is obtained by averaging the probability of successfully preparing a given target configuration of atoms over randomly-sampled initial configurations and loss processes. We numerically compute the mean success probability using realistic physical parameters [29]. We choose the survival probability during an elementary transfer or displacement operation to be $p_\nu = p_\alpha = 0.985$, and the survival probability during an elementary transfer or displacement no-op operation to be $p_\alpha^0 = e^{-t_\alpha/\tau}$ and $p_\nu^0 = e^{-t_\nu/\tau}$, respectively, where the duration of the control operations, $t_\alpha = 15$ μs and $t_\nu = 67$ μs, and trapping lifetime, $\tau = 60$ s, have been chosen conservatively. In the following, we assume the mean loading efficiency to be $\epsilon = 0.60$ and sample over a thousand initial configurations for each simulated data point.

Because the red-rec algorithm solves atom reconfiguration problems on grids by reducing them to a sequence of reconfiguration problems on chains, we first quantify the performance of the exact 1D reconfiguration algorithm in the presence of loss, and then quantify the performance of the red-rec algorithm in the presence of loss.

A. Preparing 1D configurations of atoms in the presence of loss

We focus on the problem of preparing a compact chain of N_a^T atoms at the center of a chain of $N_t = \eta N_a^T$ static traps, where $\eta = N_t/N_a^T$ is the overhead factor. We use an exact 1D reconfiguration algorithm (see App. B), which is exact in the sense that it minimizes both the total number of displacement operations and the total number of transfer operations (a single EDI cycle simultaneously performed on all atoms); however, there is no guarantee that the total number of displacement and transfer operations will be globally minimized across multiple reconfiguration cycles in the presence of loss.

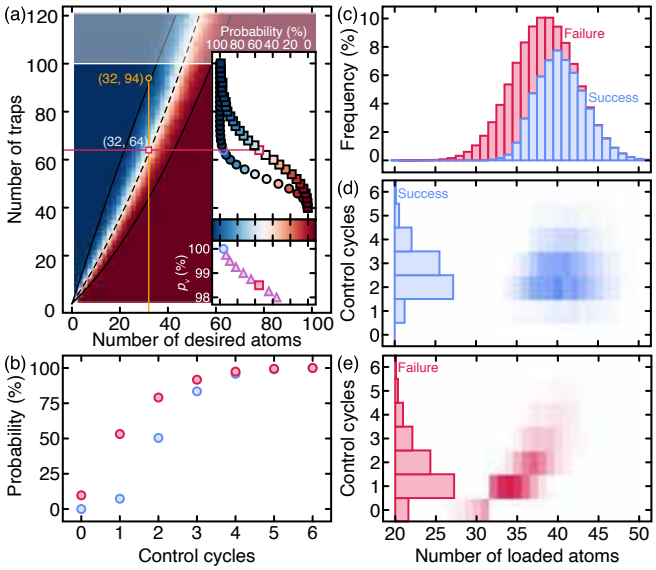

FIG. 4. **Preparing 1D configurations of atoms in the presence of loss.** (a) Mean success probability of preparing a centered-compact configuration of atoms in a 1D chain of traps in the presence of loss. The regions of near-certain failure ($\bar{p} \leq 0.02$, red) and near-certain success ($\bar{p} \geq 0.98$, blue) are separated by a sharp transition region ($\bar{p} = 0.5$, dashed curve). The number of traps needed to enter and exit the transition region increases quadratically with the number of atoms (black curves). Inset: Mean success probability of preparing a chain of 32 atoms in the absence (disks) and presence (squares) of loss (top), as well as for a varying survival probability during control operations (bottom). (b) Cumulative distribution function of the number of control cycles performed during successful (blue) and unsuccessful (red) control protocols for preparing a chain of 32 atoms in 64 traps. (c) Stacked distributions of the number of atoms loaded in the static trap array for successful (blue) and unsuccessful (red) reconfiguration protocols. (d-e) Distribution of the number of control cycles and the number of atoms loaded in the static array for successful (top, blue) and unsuccessful (bottom, red) control protocols. Integrating over control cycles returns the stacked distributions of (c).

The mean success probability surface obtained in the presence of loss (Fig. 4a) shows the same two regions of near-certain success ($\bar{p} \geq 0.98$) and near-certain failure ($\bar{p} \leq 0.02$) as in the lossless case (Fig. 3a-b) with a sharp transition region whose boundary is determined by the loading efficiency and loss parameters. In contrast to the lossless case, however, the boundary curve is non-linear, indicating that an increasingly large overhead factor is required to prepare increasingly large chains of atoms with near-certain success. We find that the overhead ratio when $\bar{p} \approx 0.50$ is well approximated by $\eta(N_a^T) \approx \eta_0 + \eta_1 N_a^T$, where $\eta_0 = 1.50(3), \eta_1 = 0.014(3)$, such that $N_t = \eta_0 N_a^T + \eta_1 (N_a^T)^2 \sim \mathcal{O}((N_a^T)^2)$ static traps are required to successfully prepare a chain of N_a^T atoms. The largest configuration of atoms that can be prepared with near-certain success in a static trap array

of 100 (120) traps is 34 (42) atoms, as opposed to 50 (61) atoms in the absence of loss.

Because of its relevance for preparing 2D configurations of $32 \times 32 = 1024$ atoms, we study in more detail the problem of preparing a chain of 32 atoms in an array of 64 traps. The probability of preparing 32 atoms in 64 traps is $\bar{p} \approx 0.5$ (for $\epsilon = 0.60$) and increases to unity as the survival probability during control operations increases to unity (inset of Fig. 4a). At least 94 traps are required to prepare a compact chain of 32 atoms with near-certain success. The success probability transitions from the near-certain failure region to near-certain success region when the survival probability of control errors increases from 0.97 to 1.00 (inset of Fig. 4a). Focusing on successful and unsuccessful protocols, we observe that the mean number of control cycles is smaller for unsuccessful protocols than for successful protocols (Fig. 4b), meaning that protocols are quicker to fail than to succeed. The probability that a successful (unsuccessful) control protocol performs fewer than 2 (1) control cycles is one half. We also observe that successful protocols have more atoms on average in their initial configuration than unsuccessful protocols (Fig. 4c); the surplus atoms are used in late reconfiguration cycles to replace atoms lost in early reconfiguration cycles. The probability that a control protocol is successful given that N_a^0 atoms have been loaded into the static array is greater than one half for $N_a^0 \geq 37$. These observations are also contained in the probability distribution functions for the numbers of control cycles and the numbers of atoms in the initial configuration for successful and unsuccessful protocols (Fig. 4d-e).

B. Preparing 2D configurations of atoms in the presence of loss

We now focus on the problem of preparing a configuration of $N_a^T = \sqrt{N_a^T} \times \sqrt{N_a^T}$ atoms at the center of a square lattice of $N_t^x \times N_t^y$ static traps, where the width of the trap array is chosen to equal the width of the target configuration, $N_t^x = \sqrt{N_a^T}$. The resulting probability surface exhibits a region of near-certain failure ($\bar{p} \leq 0.98$) and a region of near-certain success ($\bar{p} \geq 0.02$) with a transition region much sharper than in the 1D case. The accessible domain is again restricted by the total number of optical traps along each direction, $N_t^{y,max} = N_t^{x,max} = 100$, and the total number of optical traps, tracing the hyperbola $N_t^y N_t^x = 2{,}048$. The accessible domain distinguishes algorithmic limitations from hardware limitations, e.g., resulting from a limited field of view or a limited number of traps.

Compared to the lossless case, the region of near-successful realizations in the presence of loss is realized with a larger numbers of traps (Fig. 5a). These additional traps are required to load the surplus atoms needed to compensate for loss during control operations. The transition curve between the failure and success regions evaluated at $\bar{p} \approx 0.5$ scales approximately linearly with the number of atoms or quadratically with the width of the trap array ($N_{t_s^x} = \sqrt{N_a^T}$). From linear regression, we estimate that $\eta(N_a^T) \approx \eta_0 + \eta_1 N_a^T$ with $\eta_0 = 0.01564, \eta_1 = 1.579$. This relationship implies that the number of traps required to assemble a configuration of N_a^T atoms scales as $N_t = N_t^x \times N_t^y \sim \mathcal{O}((N_a^T)^{3/2})$, in contrast to the linear scaling observed in the absence of loss.

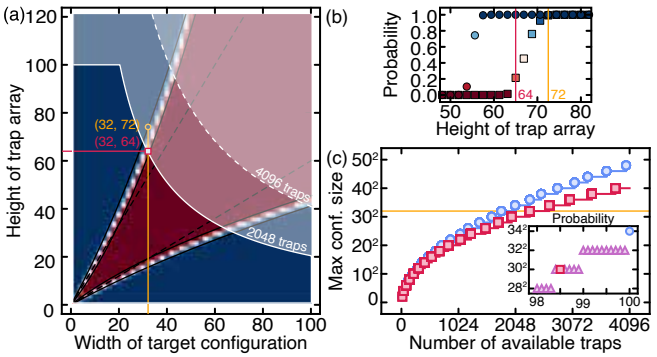

FIG. 5. **Preparing 2D configurations of atoms in the presence of loss.** (a) Mean success probability of preparing a configuration of $N_a^T = \sqrt{N_a^T} \times \sqrt{N_a^T}$ atoms in a static trap array of $N_t^x \times N_t^y$ traps with $N_t^x = \sqrt{N_a^T}$ using the red-rec algorithm in the presence of loss. The dimensions of the trap array are limited by the field of view, chosen to 100 traps along each direction, and the total number of traps chosen to 2048 traps (solid white line) and 4096 traps (dashed white line), defining an inaccessible region of parameters (shaded white region). (b) Mean success probability of preparing a configuration of 32^2 atoms in the absence (disks) and presence of loss (squares) for varying array heights. (c) Largest configuration of atoms that can be assembled with near-certain success in the absence (blue disks) and presence (red squares) of loss for a varying largest number of traps. Inset: Largest configuration of atoms that can be prepared with near-certain success given 2048 traps for increasingly efficient control operations.

The mean success probability of preparing a configuration of 32×32 atoms is limited by the total number of optical traps. Increasing the total number of optical traps from 2,048 (32×64) to 2,304 (32×72) traps increases the mean success probability from $\bar{p} = 0.21(1)$ to $\bar{p} = 0.993(2)$ (Fig. 5b). The largest configuration of atoms that can be prepared with near-certain success increases with the number of traps (Fig. 5c) and the survival probability during control operations (inset of Fig. 5c). For an array of 32×64 traps, the mean success probability increases from $\bar{p} = 0.21(1)$ to $\bar{p} = 0.996(4)$ when the survival probability during control operations is increased from $p_\nu = p_\alpha = 0.985$ to $p_\nu = p_\alpha = 0.990$.

To identify the conditions responsible for successful realizations of the red-rec algorithm, we compute the mean total number of displacement and transfer operations performed during each reconfiguration cycle (Fig. 6). We restrict our analysis to the problem of preparing a 2D configuration of $N_a^T = 32 \times 32$ atoms in a static array of

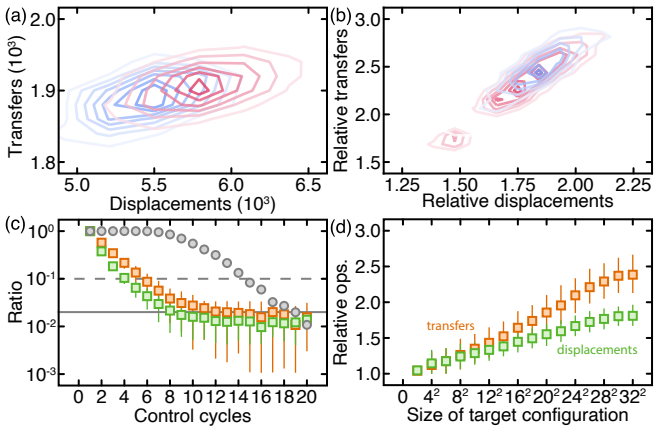

FIG. 6. **Control operations to prepare 2D configurations of atoms in the presence of loss.** (a) Distribution of the total number of transfer operations (transfers) and displacement operations (displacements) during the first reconfiguration cycle of the red-rec algorithm for successful (blue) and unsuccessful (red) realizations. The problem is to prepare a configuration of 32×32 atoms in an array of 32×64 traps. (b) Distribution of transfers and displacements during all the subsequent reconfiguration cycles expressed relative to the first reconfiguration cycle. The isolated island corresponds to initial configurations with fewer atoms than needed to successfully solve the problem in the presence of loss (the region where the blue distribution is near-zero in Fig. 7a). (c) Probability that a successful protocol will perform at least N_c control cycles (shaded disks) and total numbers of transfer (orange) and displacement (green) operations performed during each control cycle of successful realizations expressed relative to the first control cycle. Less than 10 % (2 %) of successful protocols perform more than 15 (19) reconfiguration cycles. (d) The total numbers of transfer (orange) and displacement (green) operations performed during each control cycle expressed relative to the first control cycle for increasingly large target configurations at constant overhead factor ($\eta = 2$).

$N_t = 32 \times 64$ traps.

Firstly, we observe that successful realizations perform fewer transfer and displacement operations during their first reconfiguration cycle (Fig. 6a) and more control operations in all their subsequent reconfiguration cycles than unsuccessful realizations (Fig. 6b). Secondly, we observe that the relative total number of control operations decreases in later reconfiguration cycles, i.e., fewer operations lead to less loss, and thus fewer atoms to be replaced, *ad infinitum*, so that the late cycles perform fewer control operations to replace fewer lost atoms (Fig. 6c). Thirdly, we observe that the relative numbers of total control operations increase with the size of the configuration of atoms, with increasingly more transfer operations being performed for larger configurations (Fig. 6d). We explain this increase in the relative numbers of displacement operations by the fact that in larger arrays, surplus atoms left in their original positions must travel large distances to replace lost atoms in subsequent cycles. Because more displacement operations result in greater loss, more reconfiguration cycles are required to replace the lost atoms. On the other hand, we explain the increase in the relative numbers of transfer operations by the fact that receiver columns in late reconfiguration cycles are typically paired with multiple donor columns, each pairing requiring a streamlined sequence of control operations.

VII. REDUCING MEAN WAIT TIME VIA CONFIGURATION REJECTION

Besides improving the mean success probability of preparing a given configuration of atoms, an important operational requirement is increasing the duty cycle of quantum experiments by reducing the mean wait time between two successful reconfiguration protocols. The mean success probability depends on the number of atoms in the initial configuration, both to provide enough atoms to satisfy the target configuration of atoms, and to replace lost atoms during the multiple reconfiguration cycles. Given the distribution of the number of atoms in the initial configuration partitioned for successful and unsuccessful realizations (Fig. 7a), we can identify a first threshold below which a protocol is guaranteed near-certain failure ($\bar{p} \leq 0.02$) and a second threshold below which a protocol is more likely to fail than to succeed. Rejecting configurations containing fewer atoms than any one of those two thresholds would increase the probability of successfully solving the reconfiguration problems for the retained configuration; however even though the probability of success increases with the number of atoms in the initial configuration, the probability of measuring configurations with such a large number of atoms decreases, so that rejecting these configurations monotonically decreases the probability of success (Fig. 7b). Still, because unsuccessfully solving a problem might require multiple reconfiguration cycles (Fig. 7c), and thus more time than sampling a new configuration, there exists an optimal rejection threshold (orange vertical line in Fig. 7) that minimizes the mean wait time between two successful realizations of the target configuration of atoms. This threshold is obtained by discarding as many unsuccessful configurations as possible, while rejecting as few successful configurations as possible without letting the number of measurements diverge (Fig. 7d-e).

The mean number of measurements needed to sample a configuration that contains more atoms than the threshold is $1/p(N_a^0 \geq N_a^{thresh})$, where $p(N_a^0 \geq N_a^{thresh})$ is given by the cumulative distribution function of the binomial distribution. The mean amount of time spent preparing a configuration of atoms containing more atoms than the threshold is therefore $t_{MOT} + t_{image}/p(N_a^0 \geq N_a^{thresh})$, where t_{MOT} is the time needed to cool and trap atoms in a magneto-optical trap (MOT), and t_{image} is the time needed to image the configuration

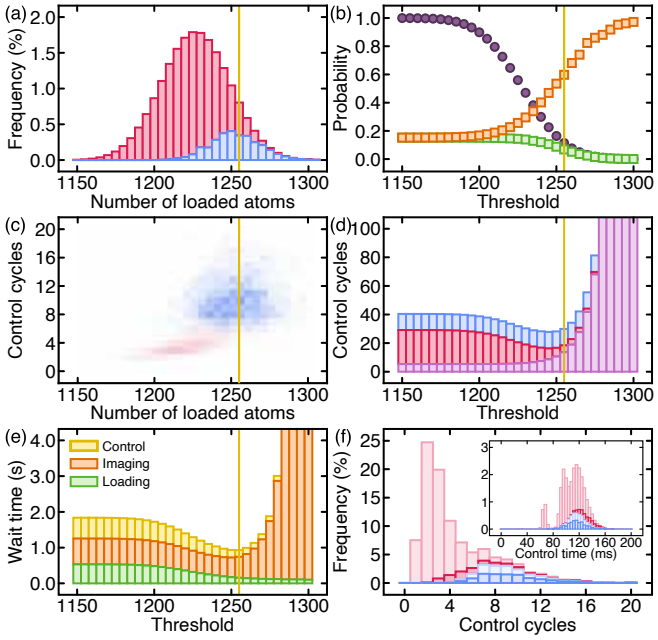

FIG. 7. **Reducing mean wait time via configuration rejection.** (a) Stacked distributions of the number of atoms loaded into a square-lattice array of $32 \times 64 = 2048$ traps for successful (blue) and unsuccessful (red) realizations at preparing a configuration of $32 \times 32 = 1{,}024$ atoms. The threshold (vertical orange line) indicates the number of atoms, N_a^*, below which configurations are rejected. (b) Probability of success given that at least N_a^* atoms have been loaded (orange circles), probability of loading at least N_a^* atoms (purple circles), and their product, which is equal to the probability of overall success (green squares). (c) Distribution of the number of control cycles per control protocol and the number of atoms loaded for successful (blue) and unsuccessful (red) realizations. Integrating over control cycles returns the stacked distributions of (a). (d) Number of measurements performed between two successful realizations for various threshold values. Measurements are performed whenever the algorithm fails to load more than N_a^* atoms (purple), or fails (red) or succeeds (blue) in preparing the target configuration given more than N_a^* atoms. (e) Stacked wait times between two successful realizations for various threshold values. The time is either spent loading the magneto-optical trap (green), imaging configurations of atoms (orange), or performing control operations (yellow). The threshold (vertical orange line) is chosen so as to minimize the total wait time. (f) Stacked distribution of control cycles for successful (stacked blue) and unsuccessful (stacked red) control protocols before (light colors) and after (dark colors) thresholding. Inset: Stacked distribution in total control time.

of atoms. We calculate the time elapsed between successful preparations of the target configuration by considering the MOT loading time, the time required to measure a configuration of atoms (imaging time), and the time required to execute the control operations. We take the MOT loading time to be $t_{MOT} = 100$ ms, the time required to image the static array to be $t_{image} = 20$ ms, and the execution time of parallel transfer and displacement operations to be $t_\alpha = 15$ μs and $t_\nu = 67$ μs, respectively. Atoms are loaded into the MOT and then continuously imaged in the static trap array until a configuration with sufficient atoms is measured; the MOT loading time is thus incurred only once for each attempted reconfiguration protocol.

From our numerical analysis, we find that the threshold value at which the minimum wait time occurs is $N_a^{thresh} = 1255$ atoms (Fig. 7f). As a result, 87.6 % of all configurations, 90.7 % of which would result in failure, are discarded. After thresholding, 61.2 % of control protocols result in success. For a configuration of $N_a^T = 32 \times 32$ atoms in a square-lattice array of size $N_t = 32 \times 64$, thresholding reduces the mean time between successful preparations of the target configuration from 1.84 s to 0.932 s. We also observe that post-thresholding, the distribution in the mean number of reconfiguration cycles and the control time for successful and unsuccessful protocols are similar (Fig. 7e), indicating that achieving success depends mostly on random realizations of atom loss rather than on the number of atoms in the initial configuration.

VIII. CONCLUSION

In conclusion, we have introduced the red-rec algorithm to solve atom reconfiguration problems on lattice geometries, specifically focusing on grids, and numerically benchmarked its performance against exact and approximation algorithms in the absence and presence of atom loss. The red-rec algorithm reduces the problem of preparing 2D configurations of atoms to the problem of solving individual columns and pairs of donor-receiver columns using efficient algorithms that exploit parallelism in hardware implementations of control operations. Individual columns are reconfigured using an exact 1D reconfiguration algorithm, whereas pairs of donor-receiver columns are solved heuristically by redistributing atoms from a donor column to a receiver column using a streamlined sequence of control operations simultaneously acting on multiple traps.

In the absence of loss, a numerical comparison of exact and approximation algorithms shows that red-rec performs well at minimizing displacement operations, but performs more transfer operations than the 3-approx algorithm. Further reduction in the number of transfer operations might be achieved by enabling redistribution of surplus atoms in the target region over both rows and columns, as well as solving for blocks of columns of size greater than two.

In the presence of loss, our results indicate that implementing the red-rec algorithm would enable preparing large configurations of atoms at fixed mean success probability and achieving fast preparation times at fixed configuration sizes, enabling the assembly of arrays of 256 and 1024 atoms using 512 and 2048 optical traps with a success probability of $\bar{p} = 0.913(2)$ and $\bar{p} = 0.21(1)$, re-

spectively. Rejecting configurations of atoms containing fewer atoms than a given threshold would result in even faster preparation times.

The red-rec algorithm is readily and efficiently implementable, and is also compatible with standard acquisition and control systems. Future work will focus on extending the red-rec algorithm to other geometries and higher dimensions, efficiently implementing it on parallel-processing devices such as graphics processing units, and experimentally quantifying its operational performance. More generally, further opportunities exist regarding the development of exact and approximation algorithms simultaneously optimizing over multiple objective functions. These algorithms might in turn guide the development of heuristic or hybrid exact-heuristic algorithms, while supporting their deployment in other domains of applications.

IX. ACKNOWLEDGEMENTS

We acknowledge early contributions from Hongru Xiang, Drimik Roy Chowdhury, Jin Tian (Benny) Huang, and Ho June (Joon) Kim. This work was supported by Industry Canada, the Canada First Research Excellence Fund (CFREF), the Canadian Excellence Research Chairs (CERC 215284) program, the Natural Sciences and Engineering Research Council of Canada (NSERC RGPIN-418579 and RGPIN-2022-02953) Discovery program, the Canadian Institute for Advanced Research (CIFAR), and the Province of Ontario. Amer E. Mouawad's work was supported by the Alexander von Humboldt Foundation and partially supported by the PHC Cedre project 2022 "PLR".

Appendix A: Exact and approximation algorithms

A reconfiguration algorithm provides a deterministic procedure that can be implemented as a sequence of programmable instructions to compute a valid solution to a given reconfiguration problem. *Exact algorithms* return solutions that are provably optimal, whereas *approximation algorithms* return approximate solutions with provable guarantees on their distance to the optimal solution. In other words, μ-approximation algorithms return solutions that are at most (at least) μ times the optimal value in the case of a minimization (maximization) problem.

We have previously discussed exact and approximation algorithms to solve reconfiguration problems on graphs from a theoretical standpoint [39]. We now review three algorithms that are relevant for the present study: (1) the exact tree algorithm, (2) the minimum-weight perfect matching algorithm, and (3) the 3-approximation algorithm.

The *exact tree algorithm* is an exact reconfiguration algorithm on trees, i.e., graphs in which any pair of vertices is connected by exactly one path, which is defined as a finite sequence of edges. An example of a tree is a one-dimensional chain. The exact tree algorithm returns a sequence of elementary displacement operations whose number is the minimum required to solve the reconfiguration problem. These operations can be further batched into a sequence of parallel control operations that minimize the number of control steps or total reconfiguration time. This algorithm is readily amenable to efficient implementation on computing hardware, including both CPUs and GPUs.

The *minimum-weight perfect matching algorithm* (MWPM) is an assignment algorithm that can be easily adapted into an exact reconfiguration algorithm valid on any graph that returns an optimal solution to the problem of minimizing the total distance travelled by atoms, i.e., the total number of displacement operations, and it does so in polynomial time. The atoms in the initial and target configurations of atoms, associated with occupied traps in the static trap array, naturally define a weighted bipartite graph upon which this problem may be solved, where the weight of an edge represents the length of a shortest path in the original graph. This algorithm does not consider transfer operations, i.e., it does not distinguish between moving one atom twice or two atoms once. Indeed, for any pair of initial and final traps, the algorithm chooses any arbitrary path among the set of all of shortest paths, irrespective on the number of atoms on that path. This algorithm could be further improved by choosing the shortest paths that minimize the number of atoms displaced.

The *3-approximation algorithm* (3-approx), which is valid for any arbitrary graph, returns a 3-approximate solution to the problem of minimizing the total number of atoms displaced; this problem is NP-hard even on grids, and is equivalent to the SAT problem [52]. Given a graph, the algorithm partitions the graph into a collection of *balanced trees*, where a balanced tree is a tree containing as many atoms as target traps. It does so by solving the U-Steiner problem [52], removing edges in order to transform the graph into a collection of balanced trees. It then applies the exact tree algorithm to each tree. This algorithm could be further improved for the total number of displacement operations by penalizing convoluted paths through trees and considering "useful" trees to be those that reduce the number of displacement operations, or more generally, the total weighted distance travelled by all atoms.

Although the MWPM and 3-approx algorithms are exact and approximate algorithms that minimize either the total displacement or transfer operations, i.e., single-objective functions, they are not optimal for any non-trivial linear combinations of such two parameters, i.e., multi-objective functions. To the best of our knowledge, there exist no known optimal or approximation algorithms on grids, graphs, or any other relevant geometries that solve reconfiguration problems for mixed-objective functions, e.g., that minimize a linear combination of the numbers of displacement operations and

transfer operations in polynomial time. In fact, we know that such algorithms do not exist unless P = NP, because minimizing the number of transfer operations is NP-hard [52], and optimizing for the mixed-objective function is a more general problem, so solving it in polynomial time would imply a polynomial-time solution for minimizing the number of transfer operations. Our proposed red-rec algorithm heuristically seeks to simultaneously minimize the total number of displacement and transfer operations.

Appendix B: Exact reconfiguration algorithm for chains

We consider the problem of preparing a configuration of N_a^0 atoms in a chain of N_t static traps, which is a special case of a tree. If the number of target traps is equal to the number of trapped atoms, i.e., $N_a^0 = N_a^T$, then there is a unique assignment of trapped atoms to target traps [62]. The number of control operations performed on traps containing atoms that is required to reach the target configuration is minimized. To reduce reconfiguration time, we apply parallel control operations to simultaneously extract, displace, and implant atoms that are not initially located in their target traps. To avoid unnecessary control errors, atoms that do not need to be displaced are not extracted into dynamic traps.

When $N_a^0 \neq N_a^T$, we can still find the set of atoms that undergo the minimum number of control operations to prepare the target configuration of atoms. We do so by solving the MWPM problem on a (complete) bipartite graph $G(U, V, E)$, where U represents the set of N_a^0 trapped atoms, V represents the set of N_a^T target traps, and E represents the edges of the graph whose weights represent the distances between source traps containing atoms and target traps. The minimum-weight, maximum cardinality matching between target traps and source traps is solved with the Linear Assignment Problem Jonker-Volgenant special algorithm (LAPJVsp) [63]. If there is a surplus of atoms, atoms located in source traps unassigned to target traps will remain in their traps, whereas if there is a deficit of atoms, a subset of target traps will remain unpopulated.

[1] C. Monroe, Quantum information processing with atoms and photons, Nature **416**, 238 (2002).
[2] P. S. Jessen, I. H. deutsch, and R. Stock, Quantum information processing with trapped neutral atoms, Quantum Information Processing **3**, 91 (2004).
[3] K.-A. B. Soderberg, N. Gemelke, and C. Chin, Ultracold molecules: vehicles to scalable quantum information processing, New Journal of Physics **11**, 055022 (2009).
[4] M. Saffman, T. G. Walker, and K. Mølmer, Quantum information with rydberg atoms, Rev. Mod. Phys. **82**, 2313 (2010).
[5] A. M. Kaufman and K.-K. Ni, Quantum science with optical tweezer arrays of ultracold atoms and molecules, Nature Physics **17**, 1324 (2021).
[6] R. P. Feynman, Simulating physics with computers, International Journal of Theoretical Physics **21**, 467 (1982).
[7] S. Lloyd, Universal quantum simulators, Science **273**, 1073 (1996).
[8] C. Gross and I. Bloch, Quantum simulations with ultracold atoms in optical lattices, Science **357**, 995 (2017).
[9] F. Schäfer, T. Fukuhara, S. Sugawa, Y. Takasu, and Y. Takahashi, Tools for quantum simulation with ultracold atoms in optical lattices, Nature Reviews Physics **2**, 411 (2020).
[10] A. Browaeys and T. Lahaye, Many-body physics with individually controlled rydberg atoms, Nature Physics **16**, 132 (2020).
[11] C. Monroe, W. C. Campbell, L.-M. Duan, Z.-X. Gong, A. V. Gorshkov, P. W. Hess, R. Islam, K. Kim, N. M. Linke, G. Pagano, P. Richerme, C. Senko, and N. Y. Yao, Programmable quantum simulations of spin systems with trapped ions, Rev. Mod. Phys. **93**, 025001 (2021).
[12] M. Morgado and S. Whitlock, Quantum simulation and computing with rydberg-interacting qubits, AVS Quantum Science **3**, 023501 (2021).
[13] A. J. Daley, I. Bloch, C. Kokail, S. Flannigan, N. Pearson, M. Troyer, and P. Zoller, Practical quantum advantage in quantum simulation, Nature **607**, 667 (2022).
[14] A. Ashkin, Acceleration and trapping of particles by radiation pressure, Phys. Rev. Lett. **24**, 156 (1970).
[15] N. Schlosser, G. Reymond, I. Protsenko, and P. Grangier, Sub-poissonian loading of single atoms in a microscopic dipole trap, Nature **411**, 1024 (2001).
[16] S. Bergamini, B. Darquié, M. Jones, L. Jacubowiez, A. Browaeys, and P. Grangier, Holographic generation of microtrap arrays for single atoms by use of a programmable phase modulator, J. Opt. Soc. Am. B **21**, 1889 (2004).
[17] E. Tervonen, A. T. Friberg, J. Westerholm, J. Turunen, and M. R. Taghizadeh, Programmable optical interconnections by multilevel synthetic acousto-optic holograms, Opt. Lett. **16**, 1274 (1991).
[18] D. W. Prather and J. N. Mait, Acousto-optic generation of two-dimensional spot arrays, Opt. Lett. **16**, 1720 (1991).
[19] J. E. Curtis, B. A. Koss, and D. G. Grier, Dynamic holographic optical tweezers, Optics Communications **207**, 169 (2002).
[20] W. J. Hossack, E. Theofanidou, J. Crain, K. Heggarty, and M. Birch, High-speed holographic optical tweezers using a ferroelectric liquid crystal microdisplay, Opt. Express **11**, 2053 (2003).
[21] M. Endres, H. Bernien, A. Keesling, H. Levine, E. R. Anschuetz, A. Krajenbrink, C. Senko, V. Vuletic, M. Greiner, and M. D. Lukin, Atom-by-atom assembly of defect-free one-dimensional cold atom arrays, Science **354**, 1024 (2016).
[22] W. Lee, H. Kim, and J. Ahn, Three-dimensional rearrangement of single atoms using actively controlled opti-

cal microtraps, Opt. Express **24**, 9816 (2016).
[23] H. Kim, W. Lee, H.-g. Lee, H. Jo, Y. Song, and J. Ahn, In situ single-atom array synthesis using dynamic holographic optical tweezers, Nature Communications **7**, 13317 (2016).
[24] J. Beugnon, C. Tuchendler, H. Marion, A. Gaëtan, Y. Miroshnychenko, Y. R. P. Sortais, A. M. Lance, M. P. A. Jones, G. Messin, A. Browaeys, and P. Grangier, Two-dimensional transport and transfer of a single atomic qubit in optical tweezers, Nature Physics **3**, 696 (2007).
[25] D. Barredo, S. de Léséleuc, V. Lienhard, T. Lahaye, and A. Browaeys, An atom-by-atom assembler of defect-free arbitrary two-dimensional atomic arrays, Science **354**, 1021–1023 (2016).
[26] D. Barredo, V. Lienhard, S. de Léséleuc, T. Lahaye, and A. Browaeys, Synthetic three-dimensional atomic structures assembled atom by atom, Nature **561**, 79 (2018).
[27] D. Ohl de Mello, D. Schäffner, J. Werkmann, T. Preuschoff, L. Kohfahl, M. Schlosser, and G. Birkl, Defect-free assembly of 2d clusters of more than 100 single-atom quantum systems, Phys. Rev. Lett. **122**, 203601 (2019).
[28] P. Scholl, M. Schuler, H. J. Williams, A. A. Eberharter, D. Barredo, K.-N. Schymik, V. Lienhard, L.-P. Henry, T. C. Lang, T. Lahaye, A. M. Läuchli, and A. Browaeys, Quantum simulation of 2d antiferromagnets with hundreds of rydberg atoms, Nature **595**, 233 (2021).
[29] S. Ebadi, T. T. Wang, H. Levine, A. Keesling, G. Semeghini, A. Omran, D. Bluvstein, R. Samajdar, H. Pichler, W. W. Ho, and et al., Quantum phases of matter on a 256-atom programmable quantum simulator, Nature **595**, 227–232 (2021).
[30] C. Sheng, J. Hou, X. He, K. Wang, R. Guo, J. Zhuang, B. Mamat, P. Xu, M. Liu, J. Wang, and M. Zhan, Defect-free arbitrary-geometry assembly of mixed-species atom arrays, Phys. Rev. Lett. **128**, 083202 (2022).
[31] K. Singh, S. Anand, A. Pocklington, J. T. Kemp, and H. Bernien, Dual-element, two-dimensional atom array with continuous-mode operation, Phys. Rev. X **12**, 011040 (2022).
[32] D. Jaksch, C. Bruder, J. I. Cirac, C. W. Gardiner, and P. Zoller, Cold bosonic atoms in optical lattices, Phys. Rev. Lett. **81**, 3108 (1998).
[33] M. Greiner, O. Mandel, T. Esslinger, T. W. Hänsch, and I. Bloch, Quantum phase transition from a superfluid to a mott insulator in a gas of ultracold atoms, Nature **415**, 39 (2002).
[34] D. S. Weiss, J. Vala, A. V. Thapliyal, S. Myrgren, U. Vazirani, and K. B. Whaley, Another way to approach zero entropy for a finite system of atoms, Phys. Rev. A **70**, 040302(R) (2004).
[35] J. Vala, A. V. Thapliyal, S. Myrgren, U. Vazirani, D. S. Weiss, and K. B. Whaley, Perfect pattern formation of neutral atoms in an addressable optical lattice, Phys. Rev. A **71**, 032324 (2005).
[36] T. Calarco, U. Dorner, P. S. Julienne, C. J. Williams, and P. Zoller, Quantum computations with atoms in optical lattices: Marker qubits and molecular interactions, Phys. Rev. A **70**, 012306 (2004).
[37] D. Bluvstein, A. Omran, H. Levine, A. Keesling, G. Semeghini, S. Ebadi, T. T. Wang, A. A. Michailidis, N. Maskara, W. W. Ho, S. Choi, M. Serbyn, M. Greiner, V. Vuletic, and M. D. Lukin, Controlling quantum many-body dynamics in driven rydberg atom arrays, Science **371**, 1355 (2021).
[38] K.-N. Schymik, V. Lienhard, D. Barredo, P. Scholl, H. Williams, A. Browaeys, and T. Lahaye, Enhanced atom-by-atom assembly of arbitrary tweezer arrays, Phys. Rev. A **102**, 063107 (2020).
[39] A. Cooper, S. Maaz, A. E. Mouawad, and N. Nishimura, Parameterized complexity of reconfiguration of atoms, in *WALCOM: Algorithms and Computation - 16th International Conference and Workshops, WALCOM 2022, Jember, Indonesia, March 24-26, 2022, Proceedings*, Lecture Notes in Computer Science, Vol. 13174, edited by P. Mutzel, M. S. Rahman, and Slamin (Springer, 2022) pp. 263–274.
[40] N. Nishimura, Introduction to reconfiguration, Algorithms **11**, 10.3390/a11040052 (2018).
[41] J. v. d. Heuvel, The complexity of change, in *Surveys in Combinatorics 2013*, London Mathematical Society Lecture Note Series, edited by S. R. Blackburn, S. Gerke, and M. Wildon (Cambridge University Press, 2013) p. 127–160.
[42] T. Ito, E. D. Demaine, N. J. A. Harvey, C. H. Papadimitriou, M. Sideri, R. Uehara, and Y. Uno, On the complexity of reconfiguration problems, Theor. Comput. Sci. **412**, 1054 (2011).
[43] N. Bousquet, A. E. Mouawad, N. Nishimura, and S. Siebertz, A survey on the parameterized complexity of the independent set and (connected) dominating set reconfiguration problems, CoRR **abs/2204.10526**, 10.48550/arXiv.2204.10526 (2022).
[44] S. Fujita, T. Nakamigawa, and T. Sakuma, Pebble exchange on graphs, Discrete Appl. Math. **184**, 139–145 (2015).
[45] T. Kato, T. Nakamigawa, and T. Sakuma, Pebble exchange group of graphs, CoRR **abs/1904.00402** (2019).
[46] S. Han, N. Stiffler, K. Bekris, and J. Yu, Efficient, high-quality stack rearrangement, IEEE Robotics and Automation Letters **PP** (2017).
[47] E. D. Demaine, S. P. Fekete, P. Keldenich, H. Meijer, and C. Scheffer, Coordinated motion planning: Reconfiguring a swarm of labeled robots with bounded stretch, SIAM Journal on Computing **48**, 1727 (2019).
[48] J. van den Berg, J. Snoeyink, M. Lin, and D. Manocha, Centralized path planning for multiple robots: Optimal decoupling into sequential plans (2009).
[49] K. Solovey and D. Halperin, k-color multi-robot motion planning, The International Journal of Robotics Research **33**, 82 (2014).
[50] R. A. Leese and S. Hurley, Methods and algorithms for radio channel assignment (2002).
[51] T. Standley and R. Korf, Complete algorithms for cooperative pathfinding problems, in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume One*, IJCAI'11 (AAAI Press, 2011) p. 668–673.
[52] G. Călinescu, A. Dumitrescu, and J. Pach, Reconfigurations in graphs and grids, SIAM Journal on Discrete Mathematics **22**, 124 (2008).
[53] H. W. Kuhn, The Hungarian method for the assignment problem, Naval research logistics quarterly **2**, 83 (1955).
[54] J. Edmonds and R. M. Karp, Theoretical improvements in algorithmic efficiency for network flow problems, Journal of the ACM (JACM) **19**, 248 (1972).

[55] W. Lee, H. Kim, and J. Ahn, Defect-free atomic array formation using the hungarian matching algorithm, Phys. Rev. A **95**, 053424 (2017).

[56] C. Sheng, J. Hou, X. He, P. Xu, K. Wang, J. Zhuang, X. Li, M. Liu, J. Wang, and M. Zhan, Efficient preparation of two-dimensional defect-free atom arrays with near-fewest sorting-atom moves, Phys. Rev. Research **3**, 023008 (2021).

[57] T. Mamee, W. Anukool, N. Thaicharoen, N. Chattrapiban, and P. Sompet, Heuristic compactness maximization algorithm for two-dimensional single-atom traps rearrangement, Journal of Physics: Conference Series **2145**, 012024 (2021).

[58] Z.-J. Tao, L.-G. Yu, P. Xu, J.-Y. Hou, X.-D. He, and M.-S. Zhan, Efficient two-dimensional defect-free dual-species atom arrays rearrangement algorithm with near-fewest atom moves, Chinese Physics Letters **39**, 083701 (2022).

[59] M. O. Brown, T. Thiele, C. Kiehl, T.-W. Hsu, and C. A. Regal, Gray-molasses optical-tweezer loading: Controlling collisions for scaling atom-array assembly, Phys. Rev. X **9**, 011057 (2019).

[60] J. Ang'ong'a, C. Huang, J. P. Covey, and B. Gadway, Gray molasses cooling of ^{39}k atoms in optical tweezers (2021).

[61] A. Jenkins, J. W. Lis, A. Senoo, W. F. McGrew, and A. M. Kaufman, Ytterbium nuclear-spin qubits in an optical tweezer array, Phys. Rev. X **12**, 021027 (2022).

[62] R. M. Karp and S. R. Li, Two special cases of the assignment problem, Discret. Math. **13**, 129 (1975).

[63] R. Jonker and A. Volgenant, A shortest augmenting path algorithm for dense and sparse linear assignment problems, Computing **38**, 325 (1987).