Goal quest for an intelligent surfer moving in a chaotic flow

Klaus M. Frahm and Dima L. Shepelyansky
Laboratoire de Physique Théorique, IRSAMC, Université de Toulouse, CNRS, UPS, 31062 Toulouse, France

(Received 26 June 2023; accepted 17 October 2023; published 9 November 2023)

We consider a model of an intelligent surfer moving on the Ulam network generated by a chaotic dynamics in the Chirikov standard map. This directed network is obtained by the Ulam method with a division of the phase space in cells of fixed size forming the nodes of a Markov chain. The goal quest for this surfer is to determine the network path from an initial node A to a final node B with minimal resistance given by the sum of inverse transition probabilities. We develop an algorithm for the intelligent surfer that allows us to perform the quest in a small number of transitions which grows only logarithmically with the network size. The optimal path search is done on a fractal intersection set formed by nodes with small Erdös numbers of the forward and inverted networks. The intelligent surfer exponentially outperforms a naive surfer who tries to minimize its phase space distance to target B. We argue that such an algorithm provides unique hints for motion control in chaotic flows.

DOI: 10.1103/PhysRevE.108.054212

I. INTRODUCTION

The time evolution of probability of chaotic map dynamics in a continuous phase space is described by the Perron-Frobenius operator (see, e.g., Refs. [1,2]). In 1960, Ulam proposed a method [3] which gives a discrete finite cell approximate of the Perron-Frobenius operator for a completely chaotic map. In this method, known as the Ulam method, the transition probabilities from one cell to the others are obtained via an ensemble of trajectories generating the probabilities of Markov transitions between cells in one map iteration. This gives a finite-size Markov matrix of transitions on the corresponding Ulam network. For one-dimensional (1D) completely chaotic maps, the convergence of the discrete cell description of this Ulam approximate of the Perron-Frobenius operator (UPFO) to the continuous chaotic dynamics has been rigorously proven in Ref. [4]. The properties of UPFO were analyzed in Refs. [5–7] and [8–11], respectively, for 1D and 2D chaotic maps.

The finite cell size of UPFO effectively introduces a finite noise in dynamical equations with an amplitude proportional to a cell size. For generic symplectic maps with divided phase space, containing chaotic components and integrable islands like the Chirikov standard map [12], such a noise destroys invariant Kolmogorov-Arnold-Moser (KAM) curves [1,12] and the original Ulam method does not provide a correct description of dynamics in such cases. In Refs. [13,14], it was shown that a generalized Ulam method resolves the above problem and provides a correct transition description for the chaotic component bounded by original KAM curves. In this generalized method, the Markov transitions are obtained with specific trajectories starting only inside one chaotic component, thus generating Markov transitions only between cells of the same chaotic component. For such a case, it was established [13,14] that the spectrum of the finite-size UPFO matrix converges to a limiting density as the cell size approaches zero.

In Ref. [15], it was shown that the Ulam networks generated by the generalized Ulam method have the properties of small-world networks similar to the *six degrees of separation* which appear in social networks of people [16], actors, power grids, biological, and other networks [17–19]. Thus, as demonstrated in Ref. [15], for the Ulam networks of symplectic maps the number of degrees of separation, or the Erdös number [20], grows only logarithmically with the network size for the regime of strong chaos. This growth is related to the Kolmogorov-Sinai entropy and the positive Lyapunov exponent of chaotic dynamics [1,2,12], which leads to an exponential divergence of initially nearby trajectories. Thus, even in Ulam networks of huge size, the Erdös number remains rather moderate and any cell can be reached in about ten or several tens of transitions by a random surfer.

The concept of a random surfer is broadly used for the construction of the Google matrix G of various directed networks and is at the foundations of the Google search engine [21–23]. Such a surfer jumps randomly following cell links to other cells according to the probabilities of Markov transitions. Due to a damping factor of the Google matrix and in a case of dangling nodes (or cells), a jump can go to any other cell but the related transition probabilities are very small in comparison to the probabilities of direct links. In the case of Ulam networks of symplectic maps, e.g., the Chirikov standard map, there are no dangling nodes, neither a damping factor, and there is only a relatively small number (about ten) of transitions from one cell to other cells [13–15].

Thus, for the Ulam networks of chaotic symplectic maps, it is interesting to consider not only a random surfer but also to analyze a behavior of an intelligent surfer who has a goal quest starting from an initial node A to reach another node B following a path with the highest probability of jumps and a minimal number of jumps. So, in this paper we analyze the optimal strategies and algorithms to be followed by an intelligent surfer resolving the goal quest and moving on such Ulam networks.

We note that the problem of shortest path detection has been studied actively in computer science for various types of networks (see, e.g., Refs. [24–27] and references therein).

However, here we have a rather specific case of the Ulam networks obtained from a chaotic symplectic dynamics and we think that our analysis provides a useful view on the properties of chaos. Also, we consider a somewhat different problem where not the path length as such but a different quantity, being the *resistance*, is minimized.

The paper is organized as follows: Sections II and III present the main properties of the Chirikov standard map, the construction of the corresponding Ulam network, and also the definition of the *resistance* of a given network trajectory. Section IV describes an efficient algorithm for an intelligent surfer to determine optimal trajectories with minimal resistance between two given nodes using a specific intersection set S_I of nodes with small Erdös and inverse Erdös numbers. In Sec. V, numerical results based on this algorithm are presented, in Sec. VI the fractal structure of the set S_I is studied, and in Sec. VII the dependence of the minimal resistance on the network size is discussed. In Sec. VIII, alternative model of a naive surfer is introduced and studied, and the discussion of the results is presented in Sec. IX. Additional data and figures are presented in the Appendix.

II. CHIRIKOV STANDARD MAP

We consider the Ulam network for the Chirikov standard map [12]. This map captures the important generic features of chaotic Hamiltonian dynamics and finds a variety of applications for the description of real physical systems (see, e.g., Ref. [28]). The map has the form

$$\bar{p} = p + \frac{K}{2\pi}\sin(2\pi x), \quad \bar{x} = x + \bar{p} \pmod 1). \qquad (1)$$

Here bars mark the variables after one map iteration, p, x are conjugated variables of momentum and coordinate, and K is the usual chaos parameter. We consider the dynamics to be periodic on a torus such that $0 \leqslant x \leqslant 1$, $0 \leqslant p \leqslant 1$. It has been argued that the last KAM curve, with the golden rotation number, is destroyed at critical $K_c = K_g = 0.971635406...$ [29]. A more rigorous mathematical analysis [30] proved that all KAM curves are destroyed for $K \geqslant 63/64$ while numerical computations showed that $K_c - K_g < 2.5 \times 10^{-4}$ [31]. Thus, the golden KAM curve at $K_c = K_g$ looks to be the last one to be destroyed (see also Ref. [32]). For values $K \geqslant 2.5$ considered in this paper, the dynamics is clearly globally chaotic but a certain fraction of stable nonchaotic islands is possible. The construction of the Ulam network, explained in the next section, automatically avoids such regions.

III. CONSTRUCTION OF ULAM NETWORK

The Ulam network and related UPFO for map (1) are constructed as described in Ref. [13]. For pedagogical reasons, we give a brief summary here. First, exploiting the symmetry $x \to 1 - x$ and $p \to 1 - p$, the phase space is reduced to the region $0 \leqslant x < 1$ and $0 \leqslant p < 0.5$ and then is divided into $M \times (M/2)$ cells with certain integer values M in the range $200 \leqslant M \leqslant 3200$. Here we study three values of the chaos parameter being $K = 2.5,\ 5,\ 7$. We consider one very long trajectory of 10^{12} iterations with the initial condition at $x = p = 0.1/(2\pi)$, which is in the chaotic component, and count the number of transitions $N_{i\to j}$ from a cell i to a cell j. This allows us to determine the classical transition probabilities $p_{i\to j} = N_{i\to j}/\sum_j N_{i\to j}$. The index number i associated to each cell is constructed at the same time, i.e., each time the trajectory enters a new cell, not yet visited before, the value of i is increased by one and attributed to this new cell. Depending on the value of K, it is possible that there are stable islands or other nonaccessible regions where the trajectory never enters and therefore the network size N is typically $N < M^2/2$, but for the considered K values with $K \geqslant 2.5$ the fraction of stable islands is quite modest such that $N \approx M^2/2$. The nonvisited cells due to such islands do not contribute to the Ulam network.

In practice, we perform trajectory iterations only for the largest two values $M = 3200$, $M = 2240$ (the latter not used in this paper) and apply an exact renormalization scheme to successively reduce the value of M by a factor of 2 to smaller values of M. In this paper, we limit ourselves to the cases $M = 200, 400, 800, 1600, 3200$.

Remember that the original Ulam method [3] computes the transition probabilities from one cell to other cells with many random initial conditions per cell but, for the Chirikov standard map, this procedure is less convenient since it causes an implicit coarse graining with diffusion into the stable islands or other classically nonaccessible regions which we want to avoid. The problem of transitions between cells of a chaotic component and stability island was solved in Refs. [13,14] by using the generalized Ulam method. In this approach, one or a few trajectories are launched from one cell (or few cells) of the chaotic component, thus due to ergodicity of dynamics such trajectories generate the Markov transitions only between cells inside the chaotic component without penetration inside stability islands. The detailed description of this method is given in Refs. [13,14].

The matrix $G_{ji} = p_{i\to j}$ corresponds to an Ulam approximate of a Perron-Frobenius operator (called UPFO) satisfying $G_{ji} \geqslant 0$ and the column sum normalization $\sum_j G_{ji} = 1$. In Refs. [13,14], the (complex and real) eigenvalues λ_j with $|\lambda_j| \leqslant 1$ and eigenvectors of G were analyzed and studied in detail. In the absence of absorption, which is the case in this paper, the leading eigenvalue is $\lambda_1 = 1$ and its eigenvector components give the global ergodic rather constant density of cells which are in the chaotic region of the phase space accessible from the initial trajectory. Due to ergodicity, the choice of the initial trajectory is not important if it starts inside the global chaotic component.

In Ref. [15], among other things, also the distribution of the outgoing link number N_l of a given cell i, i.e., the number of nonvanishing matrix elements $G_{ji} > 0$ for fixed i, were analyzed. In particular, it was found [15] that the maximal possible value of N_l is $N_l^{(\max)} = 2\lceil 2 + K\rceil$ and statistically mostly even values of N_l are possible (rare odd values are possible due to boundary effects and mostly for small M).

For a given trajectory $i_j \in \{1, \dots, N\}$, $j = 0, \dots, l$ between two cells $A = i_0$ and $B = i_l$, we define the *resistance* R as

$$R = \sum_{j=1}^{l} \frac{1}{p_{i_{j-1}\to i_j}}, \qquad (2)$$

representing the sum of inverse transition probabilities over each leg of the trajectory. Of course, for each transition $i_{j-1} \to i_j$, we require $p_{i_{j-1}\to i_j} > 0$, i.e., the network allows this transition with an existing link. According to (2), a trajectory, from an initial cell A to a finite cell B with a minimal resistance value R follows a path with maximal probabilities along the path and small value l of the path length. In the following, we study the properties of such optimal trajectories with minimal R values.

IV. ALGORITHM TO DETERMINE OPTIMAL TRAJECTORIES

In the following, we consider the Erdös number $N_{E,A}(C)$ (or number of degrees of separation) [19,20] representing the minimal number of links necessary to reach indirectly the specific node C via other intermediate nodes from a particular node A, also called the hub. In this context, only the existence of a link between two nodes is important and the precise value of a transition chain of probabilities is not relevant, as long as it is nonvanishing. For a given hub A, one can quite efficiently compute $N_{E,A}(C)$ for all other nodes C of the network in a recursive way. For this, one considers at a certain level $n \geqslant 0$ the set of nodes C with $N_{E,A}(C) = n$ (at initial level $n = 0$ this set only contains the hub A itself) and one determines the corresponding set at level $n + 1$ as the set of all nodes (i) which can be reached by a direct link from the level n set and (ii) which are not a member of a former set of level $\leqslant n$ (which can be rapidly verified by keeping a list of all used nodes up this level). Initially, the size of these sets grows exponentially with $\sim N_l^n$ elements where N_l is the typical or average number of outgoing links per node (with $N_l \leqslant 2\lceil 2 + K\rceil$ for Ulam networks of the Chirikov map [15]) but the exponential growth is quickly saturated once the majority of the network is occupied and the final complexity of this algorithm is $\sim N_l\, N$, since essentially for each node we have to perform N_l tests if the outgoing link is going to a new unused node or to an already occupied node.

Our aim is to determine the optimal trajectory that goes from node A to another node B and which minimizes its resistance R defined above in (2). Node A is chosen at position $(x_0, p_0) = (0.1,\ 0.1)/(2\pi)$, which is also the initial position of the unique and very long classical trajectory used to compute the Ulam network [13]. The final node B is chosen at position $(x_0, p_0) = (0.84,\ 0.4)$ for $K = 5$ and $K = 7$. However, for $K = 2.5$, we choose a slightly different initial position for B at $(x_0, p_0) = (0.92,\ 0.4)$ to assure that B is in a zone of modest value $N_{E,A}(B) \leqslant 12$, which is quite important to determine efficiently the optimal trajectories. A change of cell positions of A, B does not affect the results if A, B are taken inside a component of strong chaos (so we avoid the vicinity of integrable islands).

The minimal length $l_{\min}$ of a trajectory between A and B, in number of links, is simply per definition the Erdös number $l_{\min} = N_{E,A}(B)$ and there are no trajectories between A and B with length $l < l_{\min}$. Furthermore, for any trajectory i_j, $j = 0, 1, 2, \ldots$ with $i_0 = A$, we know that $N_{E,A}(i_j) \leqslant j$, i.e., the Erdös number of the intermediate cell i_j after j steps with respect to the initial cell A cannot be larger than the number of j steps since, simply per definition, the Erdös number is the minimal number of steps to reach cell i_j.

For practical reasons, it is necessary to limit the search of optimal trajectories to some finite maximal length $l_{\max}$ and we choose as $l_{\max} = l_{\min} + 4 = N_{E,A}(B) + 4$, i.e., we allow potentially for four additional legs in the trajectory with respect to the minimal length, assuming that the trajectories with minimal resistance values have also a length l being close to $l_{\min}$. In most cases, except for $K = 2.5$ and small values of M, this is sufficient to find the optimal trajectory. Such trajectories are therefore limited to nodes C with Erdös number below $l_{\max}$, i.e., $N_{E,A}(C) \leqslant l_{\max}$. We call the set of such nodes $S_{E,A}$.

To find practically the optimal trajectory one has to, in principle, apply a recursive search starting from the hub but this procedure has an exponential complexity $\sim l_{\max}^{N_l}$ resulting in a large number of test trajectories that cannot reach the selected end point B. To simplify this search, we note that for each trajectory $A \to B$ in the Ulam network, we also have the inverted trajectory $B \to A$ in the inverted Ulam network (with all links inverted). Therefore, we also compute the *inverse Erdös* number $N^*_{E,B}(C)$ for all nodes C as the Erdös number of the inverted Ulam network with respect to the final point B as the hub and determine the set $S^*_{E,B}$ of nodes C with $N^*_{E,B}(C) \leqslant l_{\max}$. The cells i_j, $j = 0, \ldots, l$ of each possible trajectory $A \to B$ (with length $l \leqslant l_{\max}$ and $i_0 = A, i_l = B$) belong to both sets $S_{E,A}$ and $S^*_{E,B}$ and therefore they are limited to cells belonging to the intersection $S_{E,A} \cap S^*_{E,B}$. Actually, each cell i_j of such a trajectory even satisfies a stronger condition for its iteration number j,

$$j \leqslant l - N^*_{E,B}(i_j) \leqslant l_{\max} - N^*_{E,B}(i_j), \qquad (3)$$

since the iteration number $j^* = l - j$ of the inverted trajectory obviously satisfies $N^*_{E,B}(i_j) \leqslant j^*$ in a similar way as $N_{E,A}(i_j) \leqslant j$. From the latter inequality and (3), we find that $i_j \in S_I$ where S_I is defined as the set of cells C satisfying the condition

$$N_{E,A}(C) + N^*_{E,B}(C) \leqslant l_{\max}. \qquad (4)$$

Therefore, each possible trajectory $A \to B$ of length $l \leqslant l_{\max}$ is limited to nodes of set S_I. In the following, we also call S_I the *intersection set* because it is a (smaller) subset of the intersection $S_{E,A} \cap S^*_{E,B}$ since (4) implies both conditions for subsets $S_{E,A}$ and $S^*_{E,B}$. However, we mention that the inverse is not true, implying that S_I is a strictly smaller subset of $S_{E,A} \cap S^*_{E,B}$ and not equal to the latter.

The recursive search of optimal trajectories can be greatly improved by limiting the recursion at each step to possible nodes i_j satisfying the (second) inequality (3). In Fig. 7, the number N_T of such possible trajectories is shown for the parameter choices of A, B, M, K, and $l_{\max}$ given above. Even though this value can be quite large (largest value between 10^8 and 10^9), it is typically quite modest and clearly in the range where it is comparable to the cost of computation of both sets of Erdös numbers (which has to be done in advance; a few seconds on a single processor of a modern computer for $M = 3200$).

V. TRAJECTORIES WITH MINIMAL RESISTANCE

We consider that an intelligent surfer applies the above algorithm to find the optimal short trajectory or path with minimal resistance between nodes A and B of the Ulam

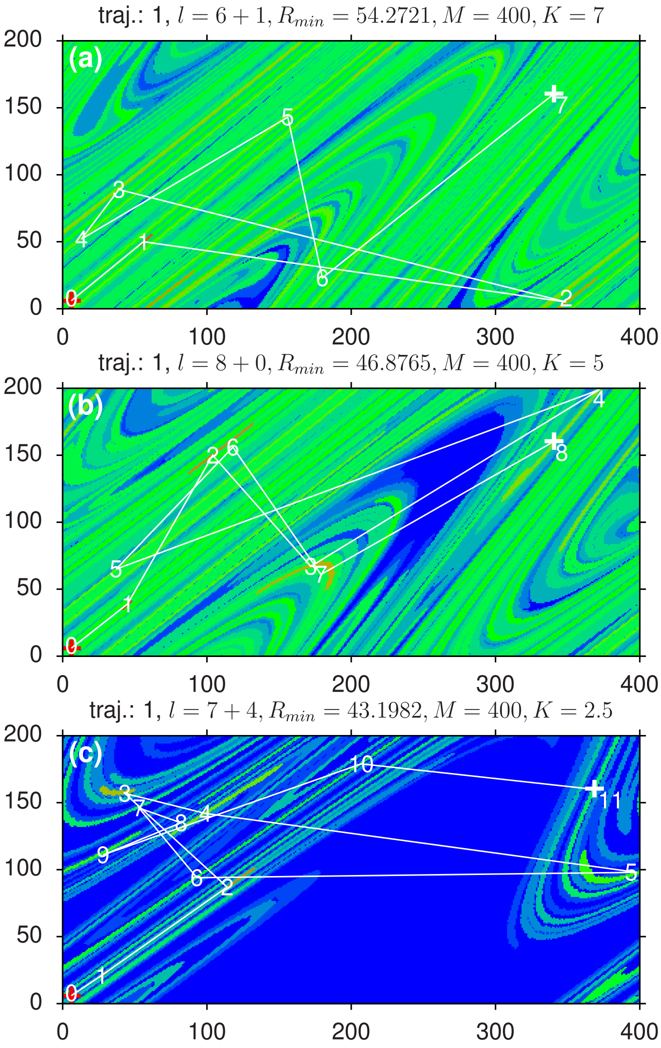

FIG. 1. The white lines show the optimal trajectory between the initial point A (red cross) and final point B (white cross) with minimal resistance R (see text for the definition). The white numbers correspond to the iteration number $j = 0, \ldots, l$ of the trajectory points i_j with l being the length of the trajectory. The color plot shows the set $S_{E,A}$ of cells C with Erdös number $N_{E,A}(C) \leqslant l_{\max} = N_{E,A}(B) + 4$ where red, green, light blue, and full blue colors correspond to maximal, intermediate, small, and negative (if $C \notin S_{E,A}$) values of the difference $l_{\max} - N_{E,A}(C)$. The values on x and y axes correspond to integer position values Mx and Mp of Ulam cells in classical phase space ($0 \leqslant x \leqslant 1$, $0 \leqslant p \leqslant 0.5$) for $M = 400$ and $K = 7$ (a), $K = 5$ (b), and $K = 2.5$ (c). The minimal resistance R and the length of the optimal trajectory are $l = N_{E,A}(B) + \Delta l$ with $R = 54.2721$, $N_{E,A}(B) = 6$, $\Delta l = 1$ (a); $R = 46.8765$, $N_{E,A}(B) = 8$, $\Delta l = 0$ (b); $R = 43.1982$, $N_{E,A}(B) = 7$, $\Delta l = 4$ (c).

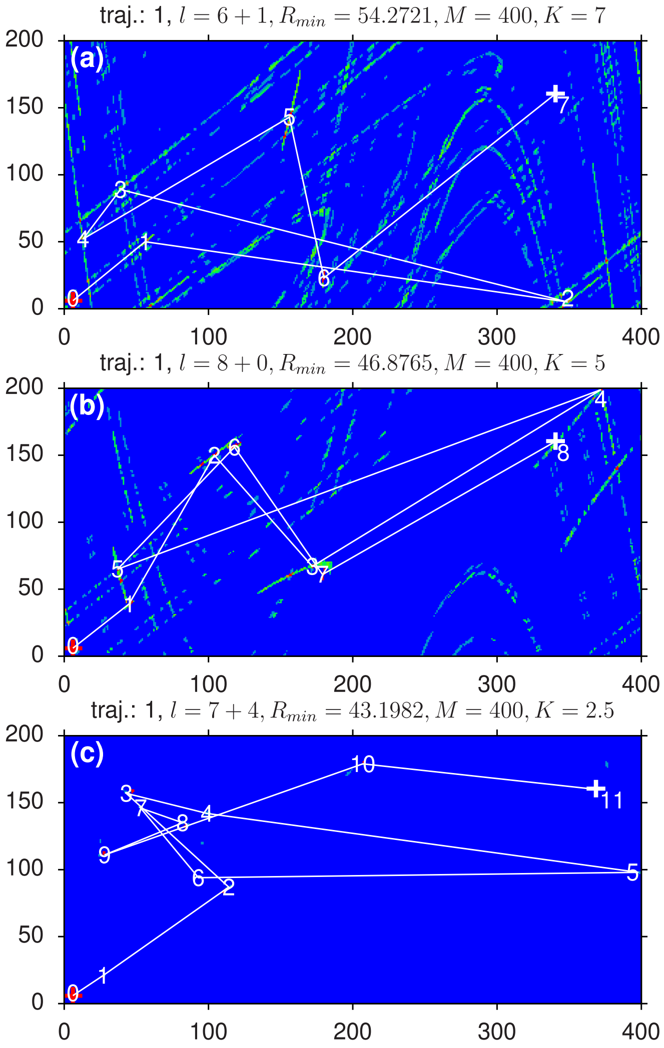

FIG. 2. As in Fig. 1 with identical trajectories and same values of K and M but with a different background color plot showing the intersection set S_I of cells C with $N_{E,A}(C) + N^*_{E,B}(C) \leqslant l_{\max}$ and where red, green, light blue, and full blue correspond to maximal, intermediate, small, and negative (if $C \notin S_I$) values of the difference $l_{\max} - N_{E,A}(C) - N^*_{E,B}(C)$.

network. Thus, using the above efficient algorithm, based on the recomputing of the Erdös numbers of inverted dynamics (with respect to B as a hub) and exploiting condition (3), he or she computes the optimal trajectories with minimal resistance (2) for networks with $200 \leqslant M \leqslant 3200$ at $K = 2.5,\ 5,\ 7$ and the initial and end points A and B given in the previous section.

In Fig. 1, we show the obtained optimal trajectories for the three K values and $M = 400$ with a background color plot representing set $S_{E,A}$ [of nodes C with Erdös numbers $N_{E,A}(C) \leqslant l_{\max} = N_{E,A}(B) + 4$]. We see that all trajectory points are on these sets and, in particular, the initial values $i_j = 1, 2, 3$ are in the limited orange regions of small Erdös numbers $N_{E,A}(C) = 1, 2, 3$. Furthermore, for smaller $K = 2.5$ ($K = 5$), set $S_{E,A}$ is strongly (modestly) reduced with respect to $K = 7$. This is explained by the reduced values of $N_l \leqslant 2\lceil 2 + K \rceil$ requiring longer trajectories to cover the same fraction of phase space. The obtained values of the minimal resistance R are of the same order of magnitude between the three K values but for smaller K they are somewhat smaller. This is due to the reduced N_l values leading to an increase of typical transition probabilities and a resistance reduction (since R is the sum of inverse transition probabilities).

Figure 2 shows the same trajectories but on a different background color plot representing the much smaller interaction set S_I defined by condition (4). One sees that according to condition (3), all trajectory points are indeed in set S_I.

Actually, depending on the iteration number j, the trajectory points (for possible trajectories between A and B) *cannot freely choose among all* points of S_I since for each value of j condition (3) represents a subset of S_I which is strictly smaller than S_I if $j < l_{\max}$.

Remember that the simple naive exponential search algorithm ensures automatically, by construction, that all search trajectories are in the (larger) set $S_{E,A}$ while the improved algorithm, exploiting condition (4), ensures that only certain search trajectories in set S_I, that can indeed go to end point B, are used. The fact that set S_I is strongly reduced in comparison to $S_{E,A}$ illustrates the efficiency of the improved search algorithm. In fact, Appendix Fig. 8 shows the number of points N_I in set S_I versus network size $N \approx M^2/2$ and we see that $N_I < 10^4$ and even $N_I \sim 10^3$ for the largest value of $M = 3200$. Thus, a strong reduction of points which belong to intersection set S_I is at the origin of the efficiency of the intelligent surfer algorithm based on relations (3) and (4).

Figure 9 is similar to Figs. 1 and 2 but using as background color plot set $S^*_{E,B}$ with limited inverse Erdös numbers (with respect to B as hub). Now, the trajectory points close to endpoint B correspond to orange regions with small inverse Erdös number. Note that S_I is a (strictly smaller) subset of the intersection $S_{E,A} \cap S^*_{E,B}$.

Figure 3 is similar to Fig. 1 (with background color plot for the set $S_{E,A}$) but for the single value $K = 7$ and four values $M = 200$, 800, 1600, 3200. The optimal trajectories for these cases are rather similar but a bit different from the optimal trajectory for $M = 400$ visible in Fig. 1(a). The fraction of nodes in set $S_{E,A}$ decreases considerably with increasing M and network size $N \approx M^2/2$, which is also due to the comparable values of $l_{\max} = 9$ ($M = 200$) or $l_{\max} = 10$ ($M \geqslant 400$).

Figure 4 is similar to Fig. 3 but with the intersection set S_I as background color plot. Here the fraction of nodes in set S_I decreases very strongly with increasing M and $N \approx M^2/2$. In particular, for $M \geqslant 1600$, only a zoomed region around the fourth node i_4 of the trajectory is shown (in full-size plots, set S_I would be essentially invisible for $M \geqslant 1600$ due to the limited resolution). Around i_4 the set of "available points" (for other trajectories from $A \to B$ with $l \leqslant l_{\text{lmax}}$) is rather well visible and of considerable size even though for larger values of M, here a zoomed representation is also necessary. The strong reduction of S_I illustrates that the improved search algorithm gains more efficiency for larger values of M. We show more results, similar to Figs. 3 and 4, at various parameters in Figs. 10–14.

We note that the above results are also valid for different positions of the initial point A or final point B inside the chaotic component. It is possible that if A or B are located in the close vicinity of stable islands, the situation may be more complex due to the known effect of sticking of trajectories in the vicinity of an integrable island (see discussions of sticking, e.g., in Refs. [13–15]). However, the number of such cells is roughly proportional to an island perimeter length and their number is relatively small even if the area of the island is significant (as for $K = 2.5$). Thus, the case when one of the A or B cells is very close to the island border should be analyzed in more detail in further studies.

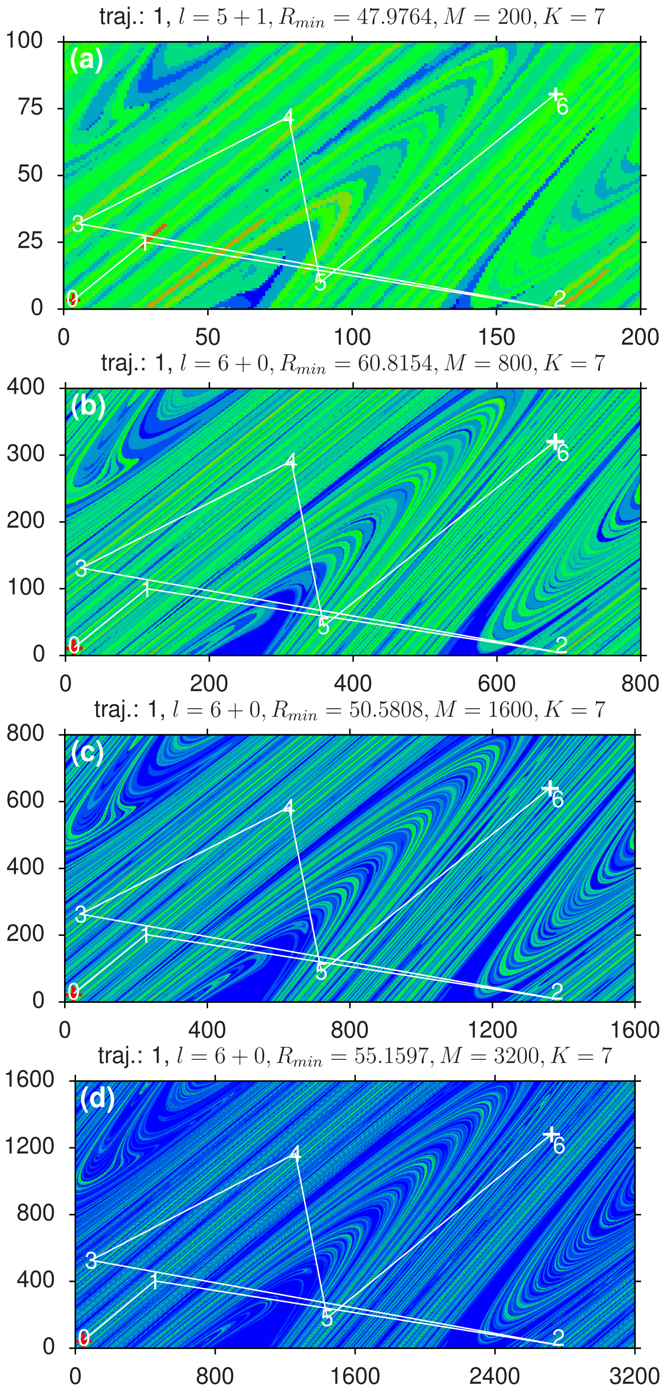

FIG. 3. As in Fig. 1 with a background color plot showing set $S_{E,A}$ for $K = 7$ and $M = 200$ (a), $M = 800$ (b), $M = 1600$ (c), and $M = 3200$ (d). The minimal resistance R and the length of the optimal trajectory are, respectively, $l = N_{E,A}(B) + \Delta l$ with $R = 47.9764$, $N_{E,A}(B) = 5$, $\Delta l = 1$ (a); $R = 60.8154$, $N_{E,A}(B) = 6$, $\Delta l = 0$ (b); $R = 50.5808$, $N_{E,A}(B) = 6$, $\Delta l = 0$ (c); $R = 55.1597$, $N_{E,A}(B) = 6$, $\Delta l = 0$ (d).

However, we mention that our algorithm also works *a priori* for such cells, but in this case the transition probabilities from or to such cells from other cells obtained by the Ulam network are rather small and we expect the resulting resistance values of optimal trajectories to be increased. This is due to the

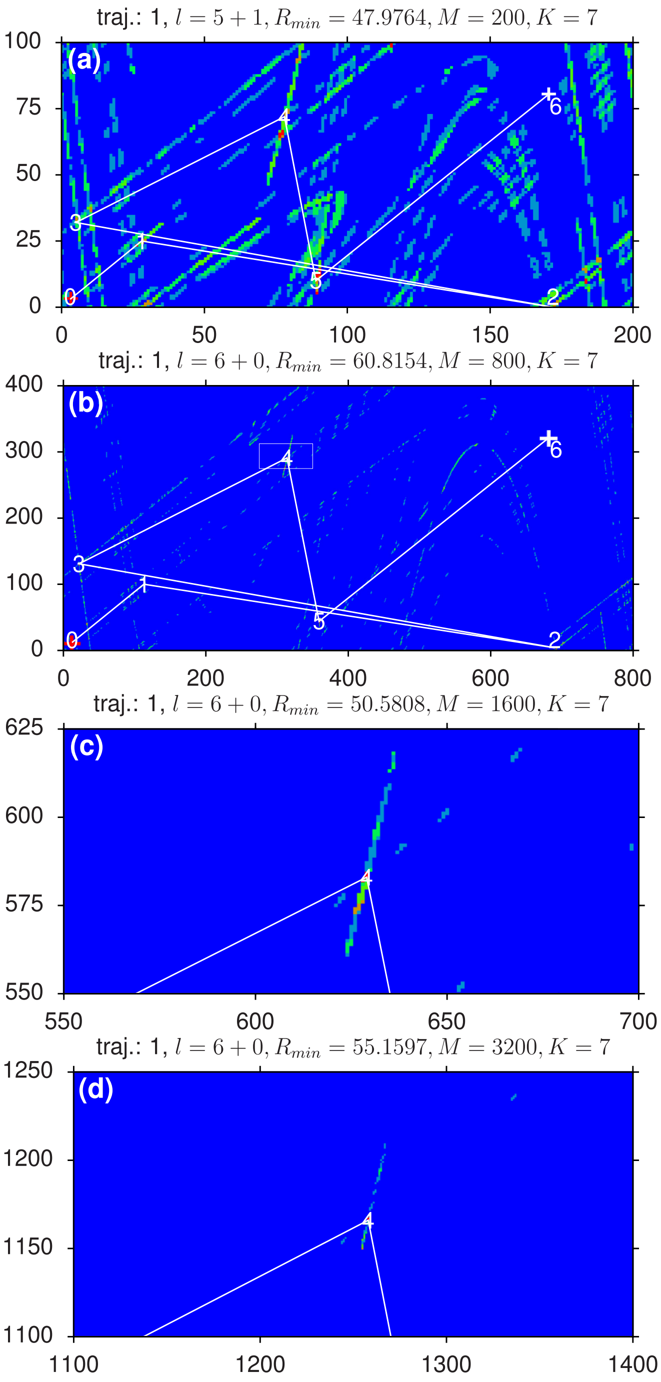

FIG. 4. As in Fig. 3 with identical trajectories and same values of K and M but with a different background color plot showing set S_I (see caption of Fig. 2). (c) ($M = 1600$) and ($M = 3200$) show a zoomed region of the phase space (around the point i_4) and corresponding to the white rectangle visible in (b) (case of $M = 800$).

fact that the Ulam network is constructed by a *single* very long trajectory in the chaotic component (accessible from the initial start cell) and generally the cells close to the stable island are only rarely visited, thus strongly reducing the corresponding transition probabilities. Furthermore, cells clearly inside the stable islands are never visited by the constructing Ulam trajectory and are, by construction not present, in the Ulam network and cannot be chosen as initial or final cells A or B.

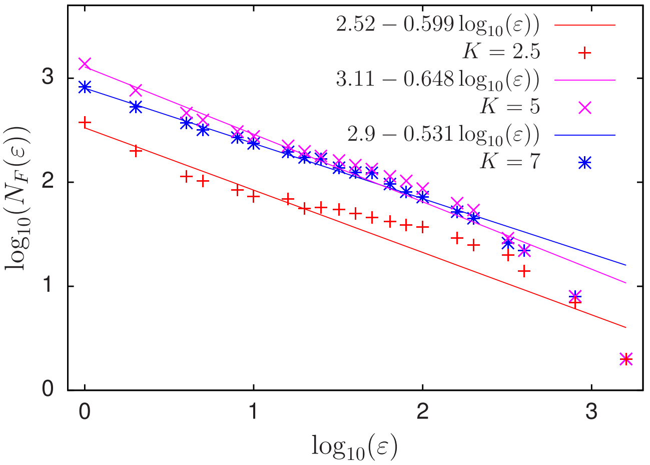

FIG. 5. Number $N_F(\varepsilon)$ of boxes of size ε required to cover set S_I versus box size ε for $M = 3200$, $K = 7$ (blue $*$), $K = 5$ (pink $\times$), and $K = 2.5$ (red $+$). The values of ε are chosen such that $\varepsilon = 1, 2, 4, 5, 10, 16, \ldots, M/2$ is a divisor of $M/2$. The straight lines correspond to the linear fit $\log_{10}(N_F(\varepsilon)) = C - D_H \log_{10}(\varepsilon)$ with $D_H = 0.531 \pm 0.013$, $C = 2.9 \pm 0.0087\,(K = 7)$, $D_H = 0.648 \pm 0.02$, $C = 3.11 \pm 0.014\,(K = 5)$, and $D_H = 0.599 \pm 0.033$, $C = 2.52 \pm 0.022\,(K = 2.5)$. Here D_H represents the box-counting fractal dimension and the fit has been done with a weight factor $w_j \sim 1/\varepsilon_j$ for the different data points $[\varepsilon_j, N_F(\varepsilon_j)]$ such that small ε values have a stronger weight.

Concerning the efficiency of our algorithm, it still depends on the Erdös number of the final cell B with respect to A (and of the inverse Erdös number of A with respect to B). It is likely that these numbers are typically larger for cells close to the stable island and then the algorithm will need more time but will still be as optimal as possible.

As far as the intermediate cells of the trajectories are concerned, it is very unlikely that they are close to the stable island since such trajectories will simply not compete for the minimal resistance.

VI. FRACTAL DIMENSION OF SET S_I

In this section, we analyze the properties of set S_I. The small size of this set is at the origin of the efficiency of the intelligent surfer algorithm discussed above. The color plots of set S_I, shown in Figs. 2 and 4 for different cases, indicate a fractal structure of this set. We therefore compute the fractal box-counting dimension D_H (see, e.g., Ref. [1] and references therein). It is determined by the behavior $N_F(\varepsilon) \sim \varepsilon^{-D_H}$, where $N_F(\varepsilon)$ is the number of boxes of size $\varepsilon \times \varepsilon$ necessary to cover set S_I (here ε takes integer values when measured in units of the Ulam grid). Figure 5 confirms the fractal behavior and shows, for $M = 3200$, the linear fits $\log_{10}(N_F(\varepsilon)) = C - D_H \log_{10}(\varepsilon)$ with $D_H = 0.531 \pm 0.013\,(K = 7)$, $D_H = 0.648 \pm 0.02$, $(K = 5)$, and $D_H = 0.599 \pm 0.033$, $(K = 2.5)$. Thus, the fractal dimension of set S_I is significantly smaller then the phase-space dimension being 2 for this 2D symplectic map. This facilitates the search for the optimal path between the two points A and B for an intelligent surfer.

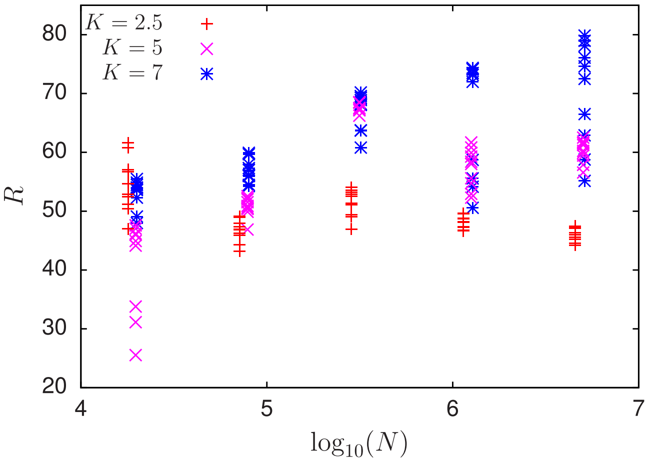

FIG. 6. Resistance R of ten optimal trajectories versus Ulam network size $N \approx M^2/2$ for $200 \leqslant M \leqslant 3200$, $K = 7$ (blue $*$), $K = 5$ (pink $\times$), and $K = 2.5$ (red $+$). The resistance is defined as the sum of the inverse transition probabilities over each leg of the trajectory from A to B [see Eq. (2)].

VII. RESISTANCE DEPENDENCE ON NETWORK SIZE

It is important to determine the dependence of the resistance R of the optimal path between A and B on the size N of the Ulam network. In Fig. 6, we show this resistance R versus $N \approx M^2/2$ for the ten best trajectories (with minimal resistance) and for all cases $2.5 \leqslant K \leqslant 7$ and $200 \leqslant M \leqslant 3200$. The dependence of $R(N)$ is not strictly monotone. For large K values ($K = 5, 7$), there is a global tendency that R is growing approximately logarithmically with N ($R \sim \log N$). This corresponds to a global logarithmic growth of the typical Erdös number with the Ulam network size N discussed in Ref. [15]. For $K = 2.5$, there are stability islands of significant size and also the Lyapunov exponent is significantly smaller in comparison to $K = 5, 7$ [12]. Due to this, we expect that higher N values are required to see asymptotic dependence $R(N)$ in this case. Also the presence of fluctuations between ten optimal orbits hides a slow logarithmic growth of $R(N)$.

We note that even for N values bigger than 1 000 000, the values of R remain relatively modest with $R \sim 60$. This value can be understood from the typical number of links per node in the Ulam network. Indeed, due to the map (1) structure, its maximal value is approximately $N_l \approx 2(2 + K) \approx 14; 18$ for $K = 5; 7$, respectively [15]. We may estimate that a typical value is smaller by a factor of 2, giving typically ten links with a typical transition probability $p_{i\to j} \sim 0.1$. According to Figs. 1–4, it takes about $l_{\text{opt}} \approx 6$ transitions for an optimal (or quasioptimal) path to go from A to B. Thus a typical resistance is $R \sim l_{\text{opt}}/p_{i\to j} \sim 60$, in agreement with the results of Fig. 6.

VIII. CASE OF NAIVE SURFER

To compare the results of the previous sections for an *intelligent surfer*, who can apply the optimal algorithm to find (i) efficiently trajectories from $A \to B$ and (ii) those with minimal resistance, we consider in this section a different model of a *naive surfer*. This naive surfer starts at the initial point $i_0 = A$ and then searches among all nodes accessible from A the cell i_1, which has the minimal Euclidean distance ($\sqrt{(p_{i_1} - p_B)^2 + (x_{i_1} - x_B)^2}$) in phase space to the target node B. Then, from this node i_1 he searches the accessible node i_2 with minimal Euclidean distance to B, etc. If at some point i_j, he can reach the final node B with one step the algorithm will automatically terminate with a found trajectory. *A priori*, one expects that the timescale for finding end point B is $\sim N$.

However, due to the birthday paradox (see, e.g., Refs. [33] and references therein), the typical timescale when this naive surfer comes back to an already visited node is $\sim\sqrt{N} \ll N$, and when this happens he enters in a periodic trajectory which will never reach end point B. Therefore, the naive surfer should also keep a list of all visited nodes and as a next step he only chooses nodes with a minimal distance to B among nonvisited nodes yet. Even in this case, there may be an "accident" when at a certain position all accessible nodes have already been visited. In this case, he goes back one step and chooses another position (with minimal distance and not yet visited).

We have implemented the algorithm of this naive surfer and it turns out that the first revisited nodes indeed happens at an iteration timescale $j_{\text{first revisited}} \sim \sqrt{N}$. Also, the second type of accident happens a few times, but in these cases only one simple back step is necessary to get out of the periodic loop. The results are given in Figs. 15 and 16.

Figure 15 shows two such trajectories for $K = 5$ and $M = 200, 400$ with colors from blue to red representing small to large iteration numbers. In both cases, a finite fraction of all available cells and nodes is used. The structure of the sets of trajectory points is quite random with no visible phase space structure [apart from the hole due to a quite big stable island at $(x, p) \approx (0.6, 0.3)$]. Figure 16 shows the length l_{naive} of these trajectories versus network size N [for all 15 values of (K, M)].

In all studied cases, l_{naive} is in the interval $[0.1 \times N, N]$, thus being of the order of network size N. Obviously, this kind of strategy to find short trajectories to end point B, without using inverse Erdös numbers, is very inefficient. Furthermore, the above algorithm of a naive surfer cannot find the trajectory with minimal resistance.

We also discuss in the Appendix (Sec. 2) the case of an intelligent surfer with limited resources and related Figs. 17 and 18.

IX. DISCUSSION

We have considered a surfer moving in a chaotic flow who is facing a goal quest to determine optimal Ulam network trajectories with minimal resistance between an initial point A and another final point B. The Ulam network is generated from the symplectic Chirikov standard map with dynamical chaos and the Perron-Frobenius eigenvector with maximal eigenvalue has ergodic probability equipartition over cells (nodes) that belong to one big connected chaotic component.

We propose an algorithm for an intelligent surfer which allows us to find a requested path of minimal resistance with a complexity $N_l^{l_{\max}/2}$, where N_l is the typical number of links per network node and $l_{\max}$ the maximal length of considered trajectories. Due to the exponential Lyapunov instability of chaotic dynamics, the number of required transitions $l \leqslant l_{\max}$

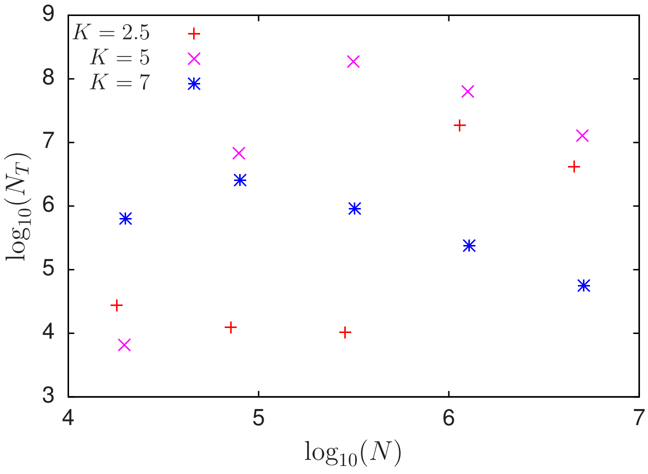

FIG. 7. Number N_T of possible trajectories from A to B with length $l \leqslant l_{\max} = N_{E,A}(B) + 4$ versus Ulam network size $N \approx M^2/2$ for $200 \leqslant M \leqslant 3200$, $K = 7$ (blue $*$), $K = 5$ (pink $\times$), and $K = 2.5$ (red $+$).

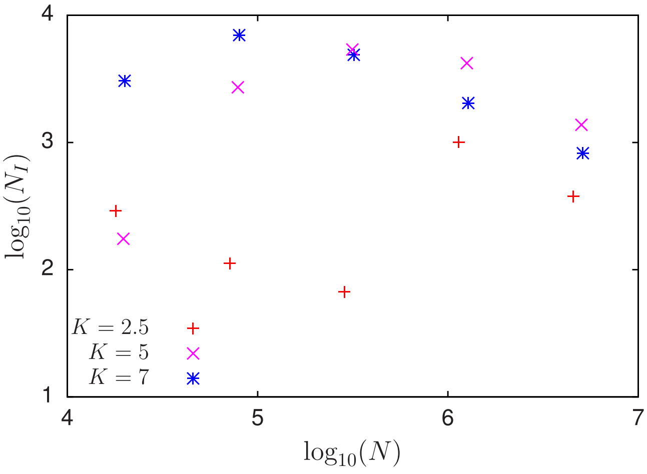

FIG. 8. Size N_I of set S_I versus the size of the Ulam network $N \approx M^2/2$ for $200 \leqslant M \leqslant 3200$, $K = 7$ (blue $*$), $K = 5$ (pink $\times$), and $K = 2.5$ (red $+$). Note that N_I corresponds to $N_F(\varepsilon = 1)$ and the three data points at $M = 3200$ [$\lg_{10}(N) \approx 6.7$] correspond to the three data points at $\log_{10}(\varepsilon) = 0$ in Fig. 5.

grows only logarithmically as $l \sim \log N$ with the network size N.

The efficiency of the algorithm is based on the computation of Erdös numbers for the directed and time-inverted dynamical flow. In particular, we have constructed the intersection set S_I of nodes satisfying condition (4) and having a fractal dimension 0.5–0.6 being significantly smaller than the phase space dimension 2.

The developed algorithm for the intelligent surfer is exponentially more efficient compared to a case of a naive surfer, who tries at each step to minimize the distance to the target point B, or to a direct recursive search algorithm without using the interaction set S_I.

We mention that most of published works and algorithms (see, e.g., Refs. [24–27]) concern typically directed networks and shortest trajectories where the iteration length l is minimized and not the resistance R (2). In our language, the computation of the minimal iteration length l is equivalent to the computation of the Erdös number. However, when a minimization of resistance is required, a more complex algorithm, based on the precomputation of inverse Erdös numbers, is needed to be used.

We hope that the algorithm of an intelligent surfer we propose will allow performing a control of motion in the regime of developed chaos in a better and more efficient way.

ACKNOWLEDGMENTS

This paper has been partially supported through Grant No. NANOX No. ANR-17-EURE-0009 in the framework of the Programme Investissements d'Avenir (project MTDINA). This paper was granted access to the HPC resources of CALMIP (Toulouse) under Allocation No. 2023-P0110.

APPENDIX

Here we present more data and figures for the main part of the paper. Thus, in Sec. A we have Figs. 7–16 and in Sec. B we have Figs. 17 and 18. The figures of Sec. A provide complementary data for the main part of the paper. Figure 7 shows the number of possible trajectories from A to B at various N and K values; also the size of set S_I is shown in Fig. 8. Figures 9–14 illustrate properties similar to those of Fig. 1–4 for various parameters. Figures 15 and 16 give additional data for the case of the naive surfer. Section B discusses the case of an intelligent surfer with limited resources with related Figs. 17 and 18.

1. Additional data and figures

Results similar to Figs. 3 and 4 are shown in Figs. 10 and 11 (for $K = 5$, 2.5 and with $S_{E,A}$ background) and Figs. 12 and 13 (for $K = 5$, 2.5 and with S_I background). For $K = 5$, it turns out that point i_4 is rather close to end point B such that the latter appears in the zoomed region for $M \geqslant 1600$. For $K = 2.5$, point i_4 of the optimal trajectory shifts considerably between $M = 800$ and $M = 1600$ such that i_4 at $M = 800$ is outside the zoomed region used for $M \geqslant 1600$. Fiure 14 of shows the second-, eighth-, and tenth-best trajectories for $M = 400$, $K = 7$ using the background color plot of set S_I. The three trajectories of Fig. 14 and also the best trajectory visible in Fig. 2(a) are somewhat different but the other six trajectories of the group of best ten trajectories are rather close to one of those four trajectories.

2. Intelligent surfer with limited resources

Ideally, an intelligent surfer could reproduce the algorithm presented above (in Sec. IV) exploiting at each search level condition (3) to avoid as early as possible all trajectories that do not link the two cells A and B. For this, he needs to compute the values of inverse Erdös numbers with respect to B as a hub at least up to level $l_{\max}$, i.e., to determine $N^*_{E,B}(C)$ for all cells C with $N^*_{E,B}(C) \leqslant l_{\max}$. In principle, he also needs the regular Erdös number $N_{E,B}(C)$ (with respect to hub A and the noninverted initial Ulam network) at least of the node $C = B$ to determine the value of $l_{\max} = N_{E,A}(B) + 4$.

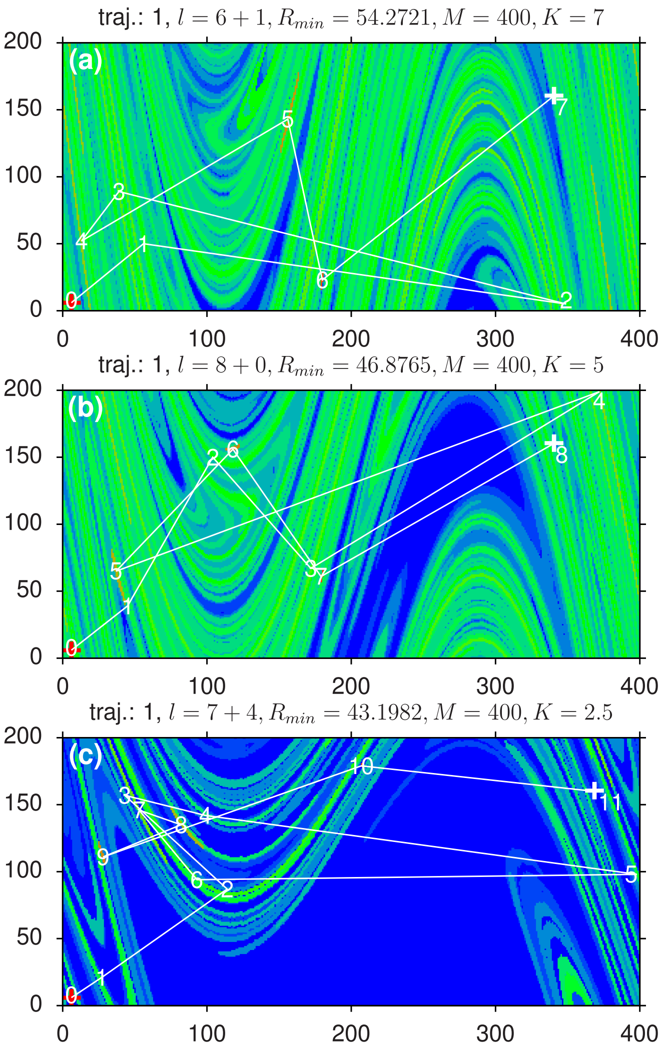

FIG. 9. As in Figs. 1 and 2 with identical trajectories and same values of K and M but with a different background color plot showing the set $S^*_{E,A}$ of cells C with $N^*_{E,B}(C) \leqslant l_{\max}$ and where red, green, light blue, and full blue correspond to maximal, intermediate, small, and negative (if $C \notin S^*_{E,A}$) values of the difference $l_{\max} - N^*_{E,B}(C)$.

Let us now assume that the surfer disposes only of limited resources (for example, in storage of Erdös numbers) and tries to minimize his initial efforts to compute both types of Erdös numbers. Concerning the computation of the value $N_{E,A}(B)$ (the only value of regular Erdös numbers which is needed), he could try to choose *ad hoc* some small value of $l_{\max}$ and apply the above algorithm, which is highly efficient especially for small values of $l_{\max}$. If $l_{\max} \geqslant N_{E,A}(B)$, he will find the nonempty set of all trajectories between A and B and therefore also the minimal length l of these trajectories which is just $N_{E,A}(B)$. If $l_{\max} < N_{E,A}(B)$, he will find no solutions and, in this case, he can increase $l_{\max}$ by one or some small value and try again.

However, to use the above algorithm he still needs to recompute the inverse Erdös numbers for a large set of nodes C up to level $l_{\max}$. The simpler and more expensive algorithm [without the test of condition (3)] becomes most expensive in the final steps where the better algorithm only uses small inverse Erdös numbers. Therefore, the intelligent surfer could decide to compute the set of inverse Erdös numbers only up to a limited depth $l_S < l_{\max}$, which is much less expensive since $N_l^{l_S} \ll N_l^{l_{\max}}$ and to apply the test of condition (3) only in the final steps when $j \geqslant l_{\max} - l_S$ which only requires us to know $N^*_{E,B}(C)$ up to level l_S.

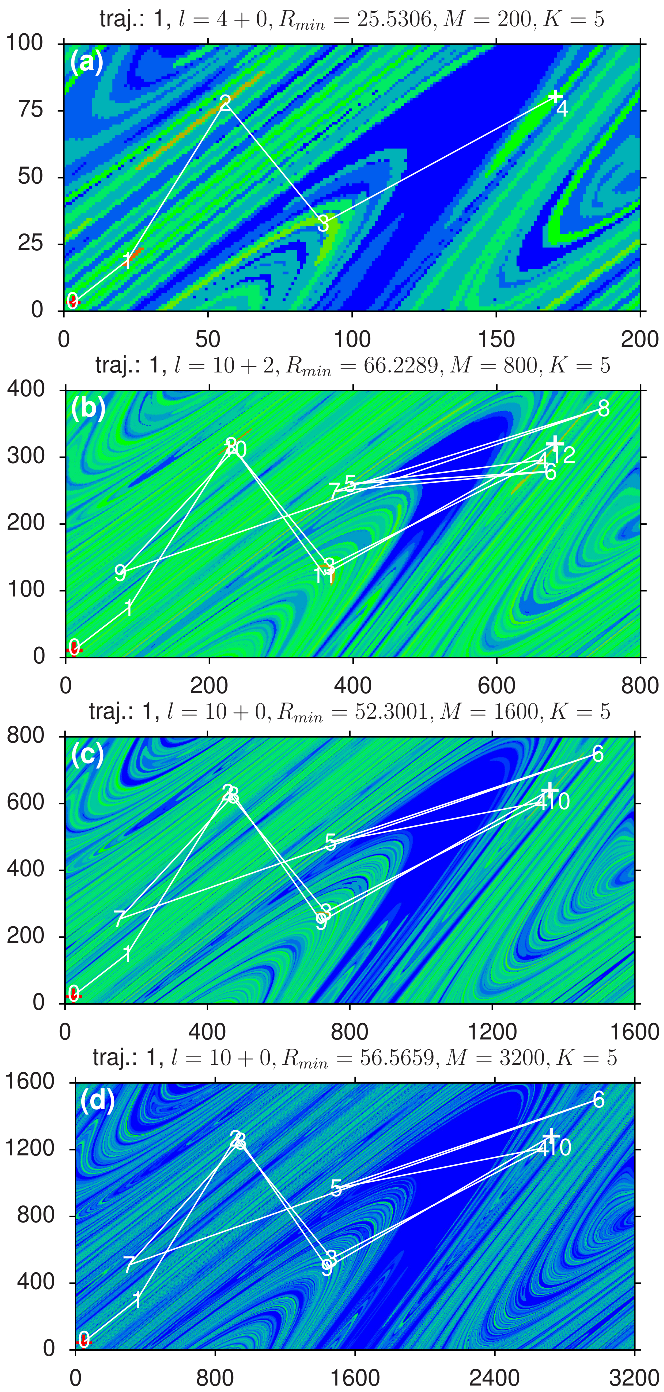

FIG. 10. As in Fig. 3 with background showing the set $S_{E,A}$ but for $K = 5$, same M values. The minimal resistance R and the length of the optimal trajectory are $l = N_{E,A}(B) + \Delta l$ with $R = 25.5306$, $N_{E,A}(B) = 4$, $\Delta l = 0$ (a); $R = 66.2289$, $N_{E,A}(B) = 10$, $\Delta l = 2$ (b); $R = 52.3001$, $N_{E,A}(B) = 10$, $\Delta l = 0$ (c); $R = 56.5659$, $N_{E,A}(B) = 10$, $\Delta l = 0$ (d).

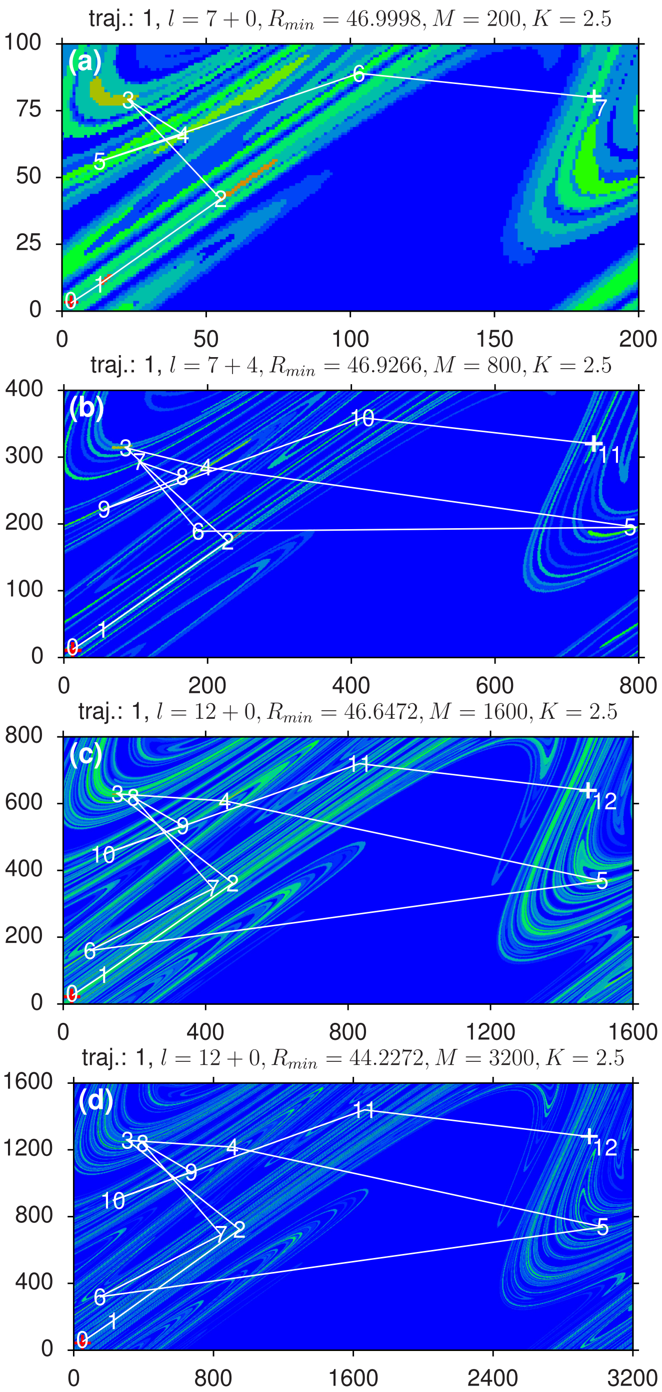

FIG. 11. As in Fig. 3 with background showing the set $S_{E,A}$ but for $K = 2.5$, same M values. The minimal resistance R and the length of the optimal trajectory are $l = N_{E,A}(B) + \Delta l$ with $R = 46.9998$, $N_{E,A}(B) = 7$, $\Delta l = 0$ (a); $R = 46.9266$, $N_{E,A}(B) = 7$, $\Delta l = 4$ (b); $R = 46.6472$, $N_{E,A}(B) = 7$, $\Delta l = 4$ (c); $R = 44.2272$, $N_{E,A}(B) = 12$, $\Delta l = 0$ (d).

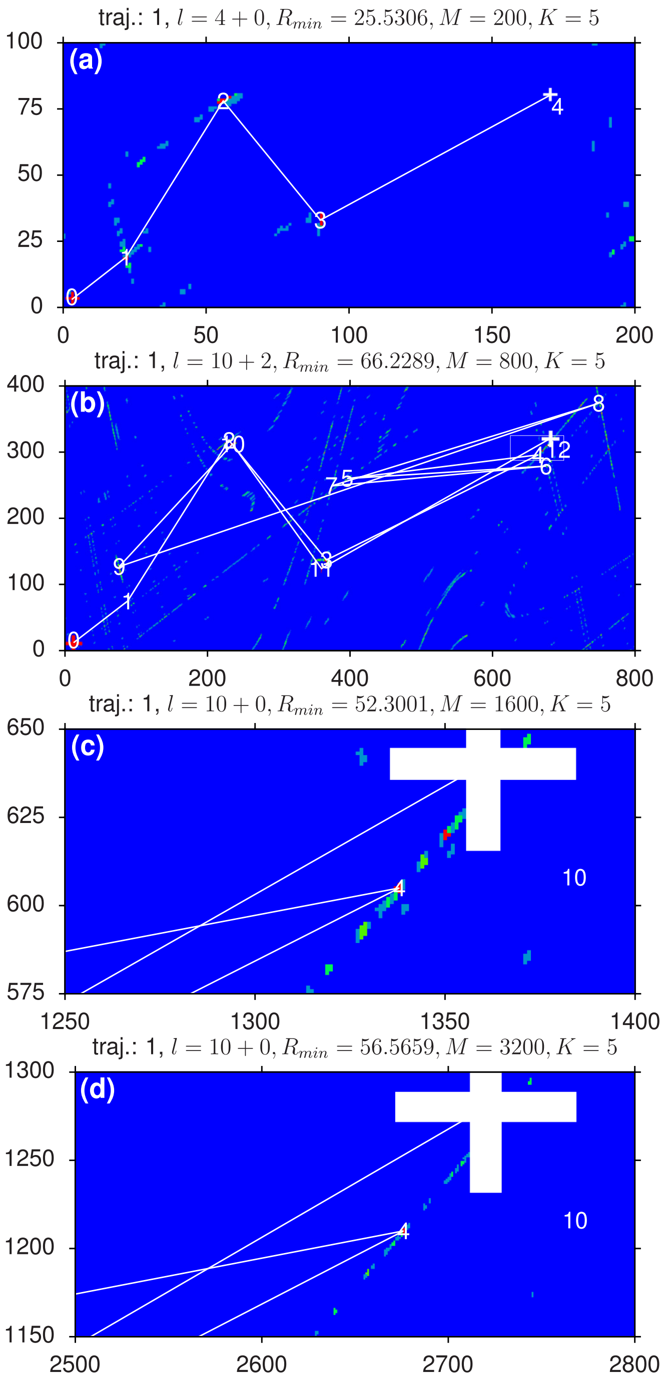

FIG. 12. As in Fig. 4 with background showing the set S_I but for $K = 5$, same M values, same trajectories of Fig. 4. As in Fig. 4, panels (c) ($M = 1600$) and ($M = 3200$) show a zoomed region (around the point i_4) of the phase space corresponding to the white rectangle visible in (b) (case of $M = 800$). The white number 10 refers to the iteration number of the visible end point $B = i_{10}$ (big white cross) which also happens to be inside the zoomed rectangle.

To measure the price and efficiency of such a simplified algorithm, we perform the recursive search in this way and compute the sum of three quantities which are (i) the number N_T of found trajectories between A and B, (ii) the number of times the recursion is stopped because condition (3) is not verified (if $j \geqslant l_{\max} - l_S$), and (iii) the number of times the recursion goes to level $l_{\max}$ but without finding a trajectory containing the final point B [case (iii) actually only happens if $l_S = 0$]. These three cases correspond to the three possibilities to stop the search recursion and their total number, which we call the *search effort number*N_S, corresponds very accurately to the computational effort. We have verified for a few

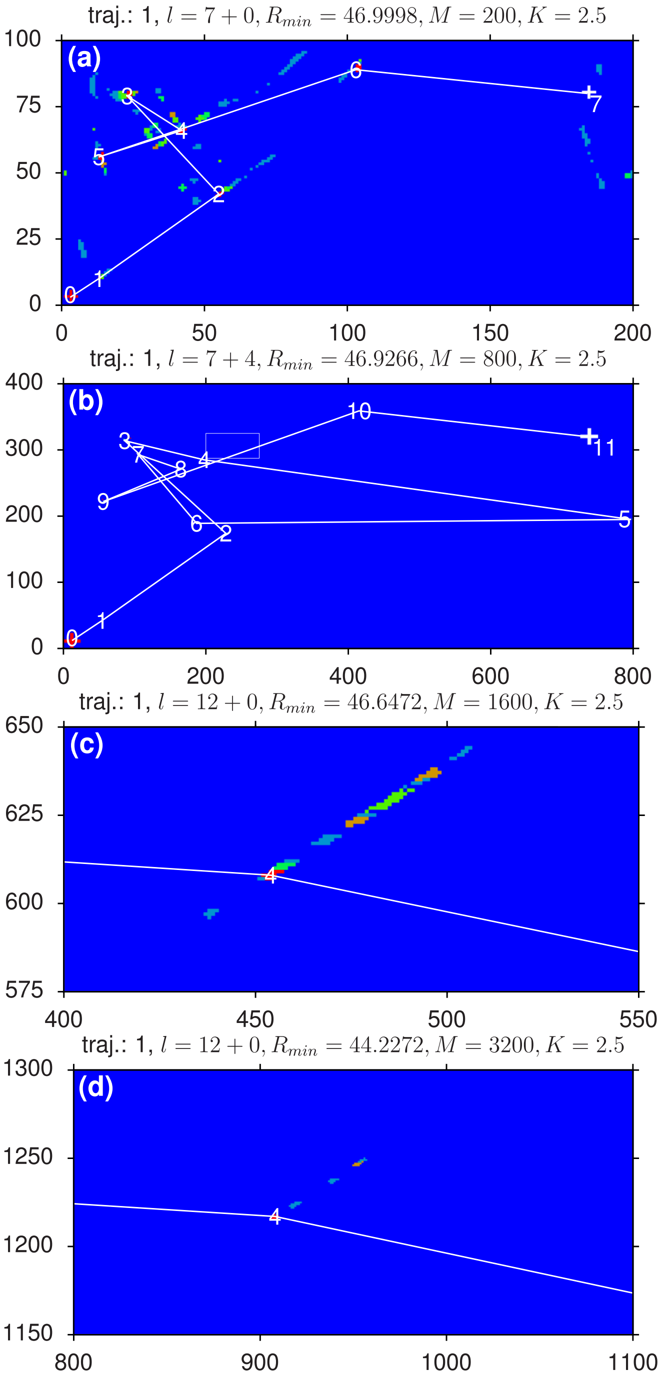

FIG. 13. As in Fig. 4, with background showing the set S_I but for $K = 2.5$, same M values, same trajectories of Fig. 11. As in Fig. 4, the (c) ($M = 1600$) and ($M = 3200$) show a zoomed region (around the point i_4 at given M value) of the phase space corresponding to the white rectangle visible in (b) (case of $M = 800$). Note that here point i_4 of the optimal trajectory at $M = 800$ (b) has a different position (outside the white rectangle) as compared to the two cases $M = 1600$ (c) and $M = 3200$ (d).

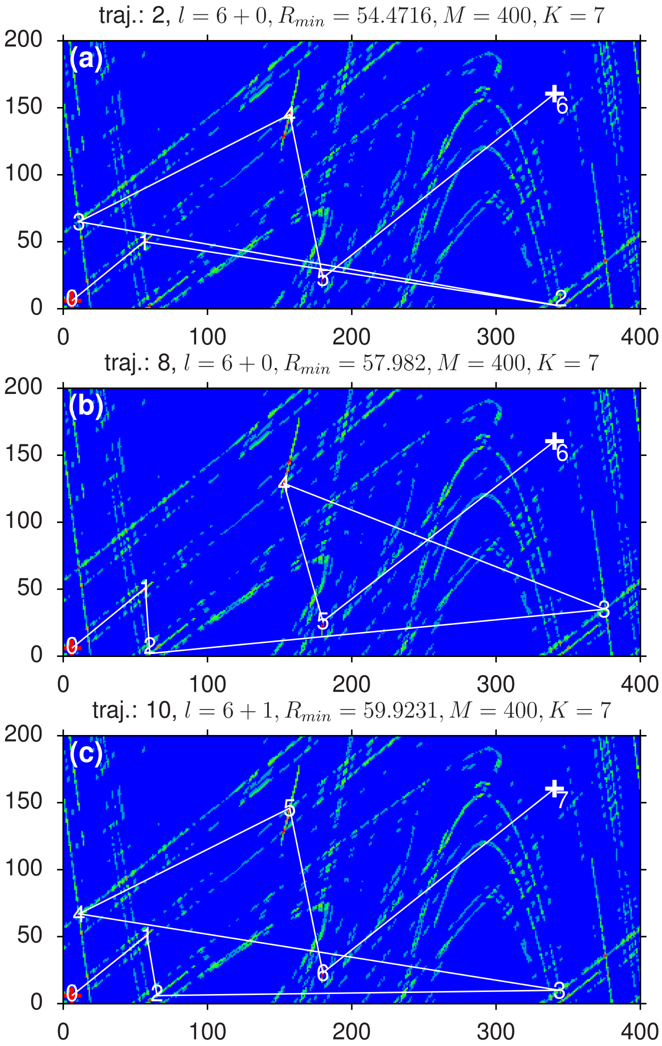

FIG. 14. As in Fig. 2, with background showing the set S_I for $K = 7$, $M = 400$, but for the second (a), eighth (b), and tenth (c) best trajectory with respect to minimal R. The minimal resistance R and the length of the trajectories are $l = N_{E,A}(B) + \Delta l$ with $R = 54.4716$, $N_{E,A}(B) = 6$, $\Delta l = 1$ (a); $R = 57.982$, $N_{E,A}(B) = 6$, $\Delta l = 0$ (b); $R = 59.9231$, $N_{E,A}(B) = 6$, $\Delta l = 0$ (c).

cases that N_S is indeed proportional to the exact computation time for different values of l_S. In the case of the perfect algorithm (with $l_S = l_{\max}$), it turns out that N_S is typically $(10 - 22) \times N_T$, which is larger than N_T but only by one order of magnitude, showing that the perfect algorithm is very efficient to find rather directly the N_T trajectories between A and B.

We have computed N_S for different values of l_S, $K = 2.5, 5, 7$ and small values of $M = 200, 400$ (for larger values of M, the cases $l_S = 0$ and $l_S = 1$ become very expensive). The result is shown in Fig. 17, showing (the logarithm) of the ratio N_S/N_T versus l_S, which starts at typical values $N_S/N_T = 10^4$–10^5 and decays very rapidly to a saturation value $N_S/N_T = 10$-22 for $l_S \geqslant 6$-7. Here the value N_T of possible trajectories for the above parameters depend on K and M and is in the range $10^3 < N_T < 10^7$ (see Fig. 7 and here the two sets of data points with $N < 10^5$ corresponding to $M = 200$ or $M = 400$, respectively). Actually, at $l_S = 3 - 4$, we already have a reduction of N_S by a factor of 100 reducing the effort to 1% showing that the strategy of the intelligent surfer to limit the value of l_S is indeed liable.

Figure 18 shows the (logarithm of the) difference $N_S - N_{S,\min}$ where $N_{S,\min}$ is the minimal value at $l_S = l_{\max}$. This

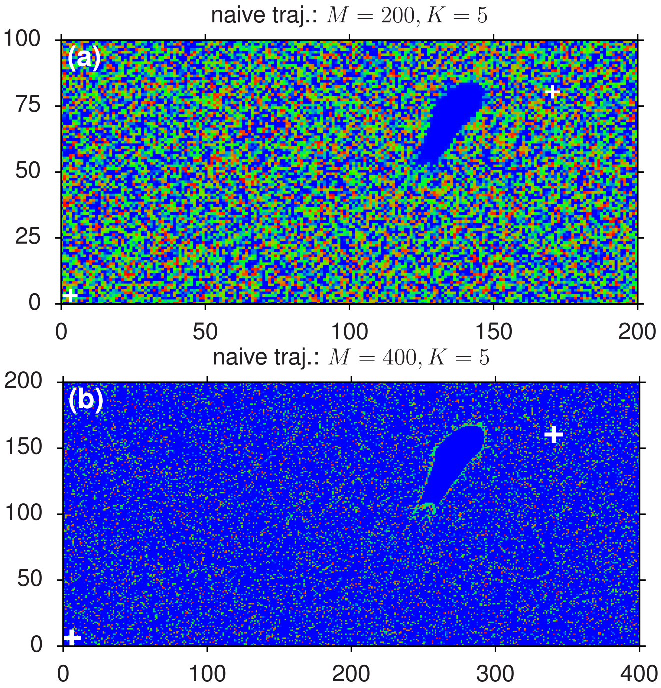

FIG. 15. Trajectory of the naive surfer for $K = 5$, $M = 200$ (a) and $M = 400$ (b). The color red (green, light blue) indicates cells with iteration numbers which are maximal (intermediate, small) in comparison to the length $l_{\rm naive}$ of the trajectory. Full blue indicate cells which are not visited by the trajectory and also stable islands whose cells are not present in the Ulam network. The white crosses show the positions of the initial cell A (bottom left corner) and the final cell (top right corner). The values of $l_{\rm naive}$ are 13982 (a) or 13744 (b), representing the fraction $l_{\rm naive}/N = 70.96\%$ (a), 17.46% (b).

representation amplifies the small differences when N_S has nearly converged to $N_{S,{\rm min}}$. At certain critical values $l_S = 8 - 12$, we have $N_S = N_{S,{\rm min}}$ and Fig. 18 does not show data points for these cases (due to the logarithm). In contrast to Fig. 17, the convergence to the limiting value seems a bit later but this is artificial due to the different presentation. The main interpretation of Fig. 18 is finally the same as in Fig. 17: At $l_S = 3\text{-}4$, there is already a strong reduction of N_S and the computational effort.

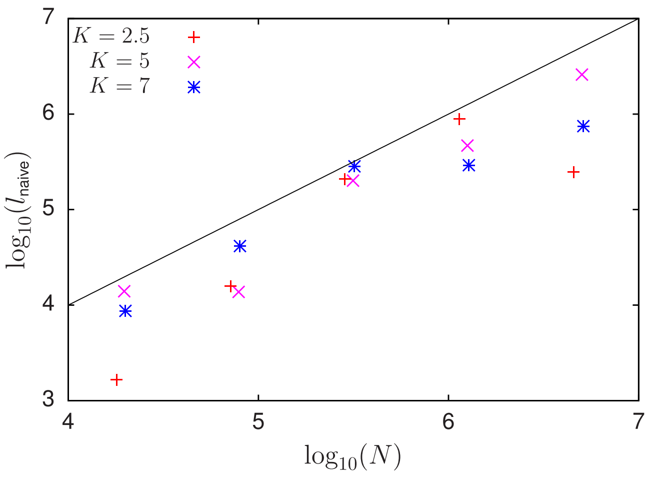

FIG. 16. Length $l_{\rm naive}$ of the trajectory of the naive surfer versus Ulam network size $N \approx M^2/2$ for $200 \leqslant M \leqslant 3200$, $K = 7$ (blue $*$), $K = 5$ (pink $\times$), and $K = 2.5$ (red $+$). The straight black line shows, for comparison, the case $l_{\rm naive} = N$.

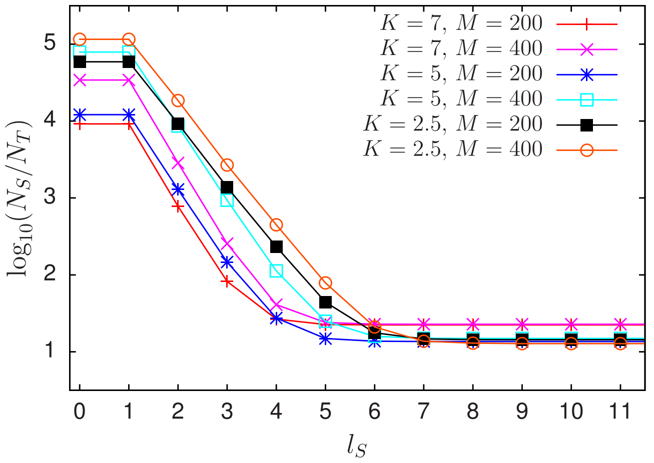

FIG. 17. Shown is the (logarithm of the) ratio of the search effort number N_S with the number N_T of possible trajectories between A and B versus search parameter l_S (see text) for $K = 2.5$, 5, 7, and $M = 200$, 400.

One could ask the question which is the optimal value of l_S to minimize the global computational cost of both precomputation of partial inverse Erdös numbers of level below l_S and the initial exponential search algorithm. The limited pre-computation of inverse Erdös numbers costs roughly $\sim N_l^{l_S}$ operations and the search algorithm needs mostly $\sim N_l^{l_{\rm max}-l_S}$ since it is essentially exponential for $l < l_{\rm max} - l_S$. The total cost

$$N_C(l_S) = N_l^{l_S} + N_l^{l_{\rm max}-l_S} \tag{A1}$$

is obviously minimal at $l_S = l_{\rm max}/2 \approx$ 5-6 for our parameters used above and close the value $l_S \approx 4$ which gives a considerable reduction of N_S according to Fig. 17.

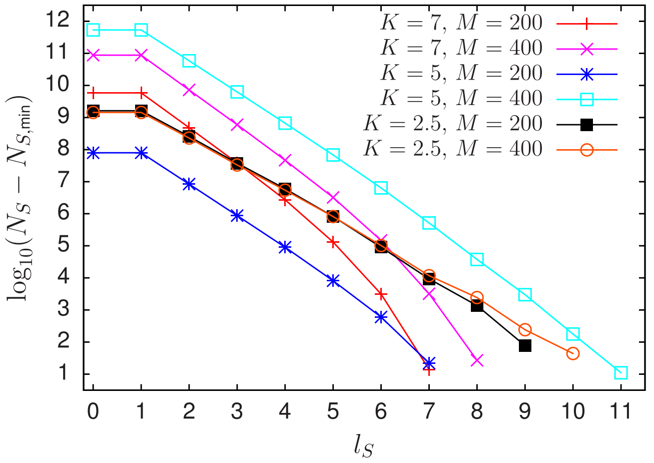

FIG. 18. Shown is the (logarithm of the) difference of the search effort number N_S with its minimal value $N_{S,{\rm min}}$ versus search parameter l_S (see text) for $K = 2.5$, 5, 7 and $M = 200$, 400. Note that in contrast to Fig. S12, the data points where $N_S = N_{S,{\rm min}}$, for sufficiently large l_S, are not visible since the logarithm of zero is not defined.

[1] A. J. Lichtenberg and M. A. Lieberman, *Regular and Chaotic Dynamics* (Springer, Berlin, 1992).

[2] P. Cvitanovic, R. Artuso, R. Mainieri, G. Tanner, G. Vattay, N. Whelan, and A. Wirsba, Chaos: Classical and quantum, http://chaosbook.org/ (accessed May, 2023).

[3] S. M. Ulam, *A Collection of Mathematical Problems*, Interscience Tracks in Pure and Applied Mathematics, Vol. 8 (Interscience, New York, 1960), p. 73.

[4] T.-Y. Li, Finite approximation for the Perron- Frobenius operator, a solution to Ulam's conjecture, J. Approximation Theory **17**, 177 (1976).

[5] Z. Kovács and T. Tél, Scaling in multifractals: Discretization of an eigenvalue problem, Phys. Rev. A **40**, 4641 (1989).

[6] Z. Kaufmann, H. Lustfeld, and J. Bene, Eigenvalue spectrum of the Frobenius-Perron operator near intermittency, Phys. Rev. E **53**, 1416 (1996).

[7] G. Froyland, R. Murray, and D. Terhesiu, Efficient computation of topological entropy, pressure, conformal measures, and equilibrium states in one dimension, Phys. Rev. E **76**, 036702 (2007).

[8] J. Ding and A. Zhou, Finite approximations of Frobenius-Perron operators: A solution of Ulamâs conjecture to multidimensional transformations, Physica D **92**, 61 (1996).

[9] M. Blank, G. Keller, and C. Liverani, Ruelle-Perron-Frobenius spectrum for Anosov maps, Nonlinearity **15**, 1905 (2002).

[10] D. Terhesiu and G. Froyland, Rigorous numerical approximation of Ruelle–Perron–Frobenius operators and topological pressure of expanding maps, Nonlinearity **21**, 1953 (2008).

[11] G. Froyland, S. Lloyd, and A. Quas, Coherent structures and isolated spectrum for Perron–Frobenius cocycles, Ergod. Th. Dynam. Sys. **30**, 729 (2010).

[12] B. V. Chirikov, A universal instability of many-dimensional oscillator systems, Phys. Rep. **52**, 263 (1979).

[13] K. M. Frahm and D. L. Shepelyansky, Ulam method for the Chirikov standard map, Eur. Phys. J. B **76**, 57 (2010).

[14] K. M. Frahm and D. L. Shepelyansky, Poincaré recurrences and Ulam method for the Chirikov standard map, Eur. Phys. J. B **86**, 322 (2013).

[15] K. M. Frahm and D. L. Shepelyansky, Small world of Ulam networks for chaotic Hamiltonian dynamics, Phys. Rev. E **98**, 032205 (2018).

[16] S. Milgram, The small-world problem, Psychol. Today **1**, 61 (1967).

[17] D. J. Watts and S. H. Strogatz, Collective dynamics of 'small-world' networks, Nature (London) **393**, 440 (1998).

[18] M. E. J. Newman, The structure and function of complex networks, SIAM Rev. **45**, 167 (2003).

[19] S. Dorogovtsev, *Lectures on Complex Networks* (Oxford University Press, Oxford, 2010).

[20] P. Erdős and A. Rényi, On random graphs I, Publ. Math. Debrecen **6**, 290 (1959).

[21] S. Brin and L. Page, The anatomy of a large-scale hypertextual Web search engine, Computer Networks and ISDN Systems **30**, 107 (1998)

[22] A. M. Langville and C. D. Meyer, *Google's PageRank and Beyond: The Science of Search Engine Rankings* (Princeton University Press, Princeton, 2006).

[23] L. Ermann, K. M. Frahm and D. L. Shepelyansky, Google matrix analysis of directed networks, Rev. Mod. Phys. **87**, 1261 (2015).

[24] E. W. Dijkstra, A note on two problems in connexion with graphs, Numer. Math. **1**, 269 (1959).

[25] S. E. Dreyfus, An appraisal of some shortest-path algorithms, Oper. Res. **17**, 373 (1969).

[26] D. B. Johnson, Efficient algorithms for shortest paths in sparse networks, J. ACM **24**, 1 (1977).

[27] M. E. J. Newman, Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality, Phys. Rev. E **64**, 016132 (2001).

[28] B. Chirikov and D. Shepelyansky, Chirikov standard map, Scholarpedia **3**, 3550 (2008).

[29] R. S. MacKay, A renormalisation approach to invariant circles in area-preserving maps, Physica D **7**, 283 (1983).

[30] R. S. MacKay and I. C. Percival, Converse KAM: theory and practice, Commun. Math. Phys. **98**, 469 (1985).

[31] B. V. Chirikov, Critical perturbation in standard map: a better approximation, arXiv:nlin/0006021.

[32] J. D. Meiss, Symplectic maps, variational principles, and transport, Rev. Mod. Phys. **64**, 795 (1992).

[33] M. C. Borja and J. Haigh, The birthday problem, Significance **4**, 124 (2007).