Computer Physics Communications 183 (2012) 1658–1663

Contents lists available at SciVerse ScienceDirect

Computer Physics Communications

www.elsevier.com/locate/cpc

Solving extra-high-order Rubik's Cube problem by a dynamic simulated annealing

Xi Chen, Z.J. Ding *

Hefei National Laboratory for Physical Sciences at Microscale and Department of Physics, University of Science and Technology of China, Hefei, Anhui 230026, PR China

ARTICLE INFO

Article history:
Received 19 July 2011
Received in revised form 25 February 2012
Accepted 5 March 2012
Available online 9 March 2012

Keywords:
Rubik's Cube
Monte Carlo
Parallel

ABSTRACT

A Monte Carlo algorithm, dynamic simulated annealing, is developed to solve Rubik's Cube problem at any extra-high order with considerable efficiency. By designing appropriate energy function, cooling schedule and neighborhood search algorithm, a sequence of moves can select a path to decrease quickly the degree of disorder of a cube and jump out local energy minima in a simple but effective way. Different from the static simulated annealing method that adjusting the temperature parameter in Boltzmann function, we use a dynamic procedure by altering energy function expression instead. In addition, a solution of low-order cube is devised to be used for high efficient parallel programming for high-order cubes. An extra-high-order cube can then be solved in a relatively short time, which is merely proportional to the square of order. Example calculations cost 996.6 s for a 101-order on a PC, and 1877 s for a 5001-order using parallel program on a supercomputer with 8 nodes. The principle behind this feasible solution of Rubik's Cube at any high order, like the methods of partial stages, the way to design the proper energy function, the means to find a neighborhood search that matches the energy function, may be useful to other global optimization problems which avoiding tremendous local minima in energy landscape is chief task.

1. Introduction

Ever since the invention of Rubik's Cube, a 3D mechanical puzzle, by Hungarian Ernő Rubik in 1974, the problems of finding its optimal solution [1], the application of group theory [2] and the cube's physical metaphors have aroused interest among scientists. However, up to this date, all the studies are limited to low-order cubes, mostly from 3 to 11. The solutions for high-order cubes and even extra-high-order cubes have little been attempted, neither its theory nor a feasible computer solution. The main difficulty is that the number of microstates, A, of a cube increases fleetly with order, n, to an extremely large number, and the difficulty of the problem hence rockets with n. For example, even at a low order, $n = 5$, $A =$ 282,870,942,277,741,856,536,180,333,107,150,328,293,127,731,985,672,134,721,536,000,000,000,000,000 [3]. It would take infinite time for a computer to randomly search a path of moves toward the final state of full order from any initial disordered state without guiding of preferential direction. The method by observing the cube's state and applying empirical formulas in solving low-order cube is impossible to be extended to the case of high orders [4].

In this article, we have applied a fast Monte Carlo method based on simulated annealing (SA) to solve any high-order cubes by introducing appropriate energy function and improved strategy. Simulated annealing method [5] has been extensively used to a variety of global optimization problems, such as, spin glass ground state, protein substructure searching, cognitive wireless networks, etc. for finding the optimal state or stable phase of a system having the minimum energy. Different from other global optimization problems in which the ground states for the final goal are unknown, for Rubik's problem we do know the ground state and that it is unique. The problem then reduces to find the shortest path among innumerous ones to the solution. The problem is computationally as hard as other global optimization problems so long as limitation of computing resource is concerned. If not properly designed the searching of the goal can easily reach the limitation but without hitting the target.

A Rubik's Cube is analogous to a six-sided 2D Ising model by regarding a color of a facelet as a vector direction of a spin. The most ordered state of a cube corresponds to the ferromagnetic phase at zero temperature in the Ising model. Solving a cube is thus converted to searching for a path leading to the stable state of the minimum energy, once the cube's energy is defined to increase with the cube's disorder. By the algorithm the accepted trial step prefers to lower the state energy. The guided path forms a Markov chain and to decrease hugely the number of trial steps toward the goal. For the sake of convenience, here we only discuss the case of odd number order. Cubes that have even number order can be solved in a similar way. The only difference is that six 2×2 central blocks with the right color need to be restored first.

* Corresponding author.
E-mail address: zjding@ustc.edu.cn (Z.J. Ding).

0010-4655/$ – see front matter
doi:10.1016/j.cpc.2012.03.003

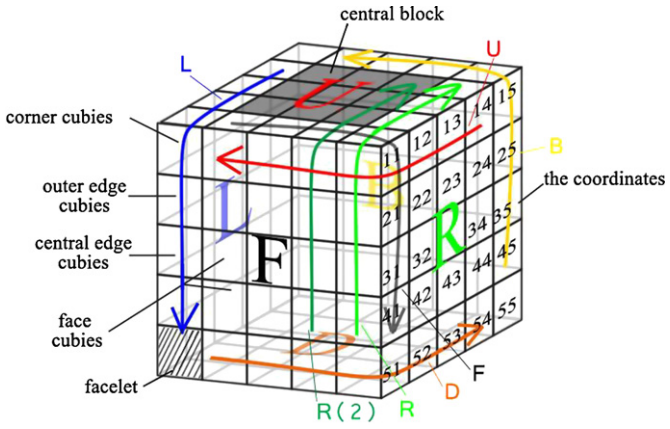

Fig. 1. Definitions of Rubik's Cube. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

2. Definition of terms and notations

A Rubik's Cube has 6 faces and each face is divided into $n \times n$ facelets. A cubie is either an edge cubie (two visible facelets, two orientations/colors), a corner cubie (three visible facelets, three orientations/colors), or a face cubie (one visible facelet, one orientation/color). A central block represents all the $(n-2) \times (n-2)$ face facelets in a specific face. An edge strip is composed of all the edge cubies of an edge. Original state is the target solution where every face of the Rubik's Cube has the unique color. In an original state, every cubie should be in one of its original positions. A face cubie has four equivalent original positions in the same face and an outer edge cubie has two. For example, as shown in Fig. 1, face cubie R22 has four original positions, R22, R24, R42 and R44. A cubie is restored when it gets to one of its original positions with the right orientation.

Six faces are expressed with capital letters, *U* (up), *D* (down), *B* (back), *F* (front), *R* (right) and *L* (left) (Fig. 1). A corner cubie is represented by three capitals, such as *UFL*. An edge strip is named similarly with two capitals, e.g. *UR*. Position of a facelet on a face is represented by two coordinate numbers, such as $R(i, j)$.

A move is defined as twisting one layer of the cube in one direction by 90°. A move is noted by the mark, "*rotation_direction*(*layer_lable*)*rotation_angle_modulus*" or "*rotation_direction*'(*layer_lable*)*rotation_angle_modulus*", where the clockwise rotations are noted by capitals and an extra apostrophe means anticlockwise twists. As for *rotation_angle_modulus*, a numeral 1 (omissible) stands for 90°- and 2 for 180°-twist. *Layer_lable* and *rotation_angle_modulus* can be omitted if they are 1. For example, $R(2)1$ or simply $R(2)$ means twisting the second layer of face-*R* clockwise by 90°, shown by the dark green arrow and the mark "$R(2)$" in Fig. 1, B' means twisting the first layer of face-*B* anticlockwise by 90°. A formula is a certain sequence of moves for reaching a given object. For example, formula $(FR'F'R)$ rotates the position of three edge strips. A trial step, which is either one move or one formula, is a movement connects the old and the new states in the process of neighborhood search.

Energy function is determined here according to the separation or the distance between the positions of two cubies in different corners, edges or faces. In geometry, a cube has 8 corners, 12 edges and 6 faces. One corner has 3 nearest corners, 3 second nearest and 1 farthest. One edge has 4 nearest edges, 6 second nearest and 1 farthest. One face has 4 nearest (proximal faces) and 1 farthest (opposite face). Then the distance *D* is defined as an integer number between two respective positions of a corner, edge or face cubie: 1 (a position to its nearest), 2 (a position to its second nearest), 3 (a position to its farthest). This definition considers the connecting path from one position to another (Fig. 2).

3. Static simulated annealing

As in other simulated annealing problems the factors that determine the computational performance of the algorithm and to be considered in the present problem are ground state, energy function, cooling schedule and neighborhood search. The energy function of a cube is considered to be only related to the current state but not history of move. The energy of a cubie is defined as a potential energy, i.e. the function of the distance between the current position and the original equivalent positions of the cubie. Two simple forms of potential energy are used here, i.e. the linear function and quadric function. The total potential energy, E, is the sum over all movable cubies:

$$E_1 = \sum_i D_i \quad \text{or} \quad E_2 = \sum_i D_i^2. \tag{1}$$

We firstly consider a static simulated annealing procedure which uses one move as a trial step and static energy function. The simplest Monte Carlo sampling is to accept a random move as a trial step only if the move lowers the total energy. It is found by a practical simulation that when the cube is in a state of enough disorder, a move hardly change the total energy; the relative energy value oscillates in a small range. The energy only drops quickly when the cube becomes quite ordered and close to its original state. The difficulty of finding energy minimum then comes not only from the presence of countless local minima in short range but also from the absence of long range energy gradient.

4. Dynamic simulated annealing

4.1. Partial stage

An improvement to the static simulated annealing is to tackle the problem in stage. By separating the whole process into certain parts and solving them one by one while keeping the solved parts unmovable, the number of states is significantly reduced from the product to the sum of the number of states in partial stages. The method is in essential to decompose the energy function into secondary energy functions, similar in spirit to Fourier analysis of complex wave shape. If the decomposition is done properly, these secondary energy functions are much smoother and simpler, more befitting for directional move. The global minimum of the total energy is obtained by finding the minima of all the secondary energy functions. According to the feature in each stage, different neighborhood search schedules are applied.

4.2. Clustering

Because of the mechanical limitation, a face cubie has only 4 positions at one face and totally 24 positions at six faces. These 24 equivalent positions are grouped to a cluster. There are $(n-1)^2/4$ clusters for all face cubies. One cubie in one cluster cannot be exchanged with other one from another cluster. This feature has vital meaning for improving computational performance. Similarly, 24 equivalent positions for the outer edge cubie compose one cluster. Thus, the project can be separated further into subproject based on clusters.

4.3. Neighborhood search

The main measure to overcome the numberless sharp and small local minima is to find appropriate formula as one trial step in every partial stage. The appropriateness of a formula can be evaluated by how much it matches with the energy function. If the value of energy function increases monotonously with number of

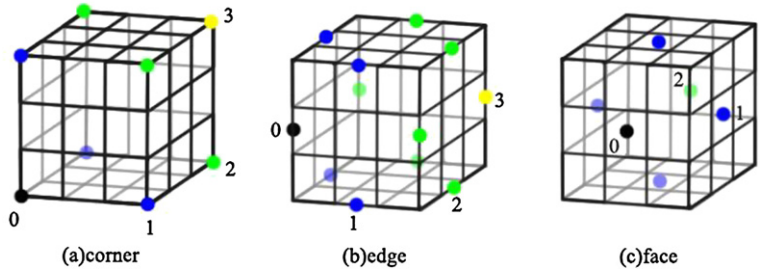

Fig. 2. Definition of distance between cubies. Black dot represents the considered cubie, its distance to blue dots is 1, to green dots is 2 and to yellow dots is 3. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

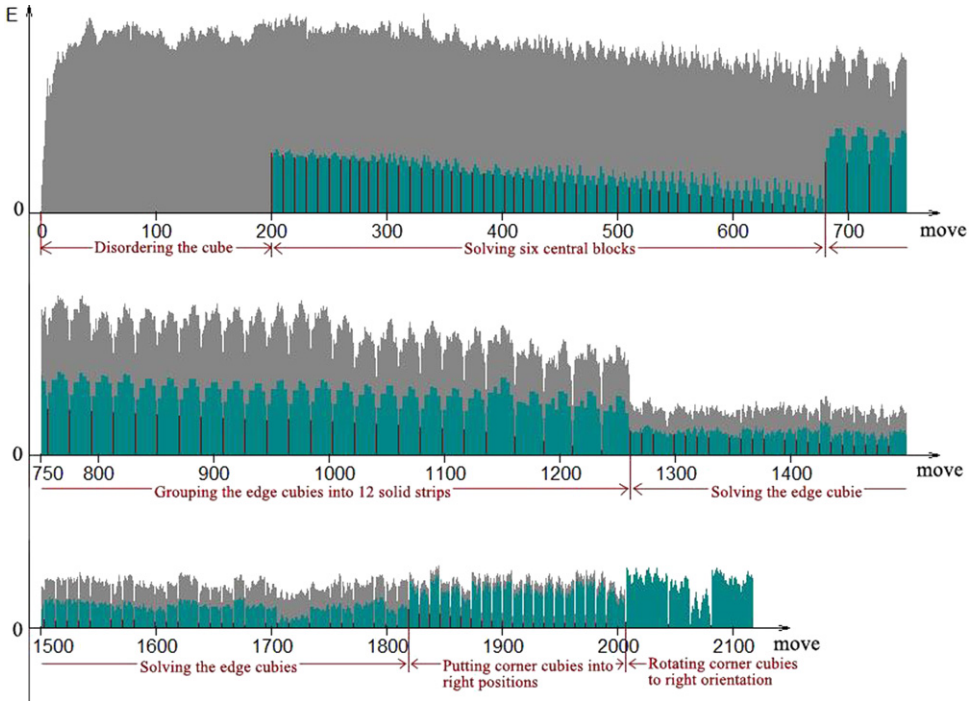

Fig. 3. The process of solving a 5-order cube using the improved SA method. The gray area shows the total energy of the cube changes with every move, while the green area shows the energy of those relevant parts of the cube in each stage plus the parts that have been solved in the previous stages. The red lines point out the last move in a formula. The figure demonstrates that by solving the cube one stage at a time, the energy image is trimmed smoother. The figure also illustrates that using one move as a trial step in neighborhood search will get into the trouble of many local minima. On the other hand, after adopting a certain formula, the states whose energy are shown by red line have good monotonicity. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

trial steps away from the original state, we say that the corresponding formula is the best matched with the energy function. The requirements for the formula are: (1) Those parts that have been restored earlier would not been destroyed. (2) Change as few cubies as possible. (3) The formula is complete. Namely every state that a cluster might possibly have can be reached by using this formula. It is hard to find a formula that satisfies all these requirements. Some formulas for low-order cubes are given by previous studies using group theory. However, to solve a Rubik's Cube at any order, formulas with extensibility need to be found. In this article, formulas are found and adapted from the Internet searching result.

Fig. 3 demonstrates an example of the energy variation, for a low-order cube, with move during a dynamic simulated annealing process in different stages. It shows clearly that a single move as a trial step in neighborhood search will highly be trapped in local minima, but using a formula as a trial step the partial energy in each step shown by red lines is monotonically reduced.

5. Specific algorithm

5.1. Solving six central blocks

In the first stage, one face facelet has four equivalent positions and its energy is only related to which face of the cube it is on. Two kinds of energy function expressions, E_1 and E_2, are used in this stage. (And the energy of edge cubies and corner cubies are neglected in this stage.) As for neighborhood search, using formula $(R(j)\ U\ L'(i)\ U'\ R'(j)\ U\ L(i)\ U')$ as a trial step [6]. This formula rotates the position of three face cubies, i.e. $F(n-j+1, i)$, $F(n-i+1, n-j+1)$ and $U(n-i+1, n-j+1)$, while not affecting other face cubies. By randomly choosing two faces as temporal face-U and face-F, the use of the formula changes the clusters to a new state. If the energy of the new state is less than or equals to old value, the trial step is accepted. Otherwise, it is rejected and a new trial process is restarted. First E_2 expression is applied, so that the cube will be twisted toward the energy minimum faster. If such trial steps are rejected for many times, namely the cube is trapped in a local minimum, then the E_1 expression is applied.

After a trial step being accepted, we return to use E_2. Hereafter the process "using the formula in a randomly chosen direction and accepting trial step that either lowers or remains the energy value" will be called *Try-out*. Do *Try-out* until all the face facelets have been restored.

5.2. Grouping the same kind of edge cubies into solid strips

In the second stage, the distance of edge cubies is defined as: $D = 1$, if an edge cubie is not the same kind as the central edge cubie. $D = 0$, if an edge cubie is of the same kind and same orientation of the central cubie. $D = 2$, if an edge cubie is the same kind (namely composed of the same two colors) as the central edge cubie but at different orientations. Because in this case, it needs to be moved away and then moved back to the edge from another direction. Formula ($L'(i)$ $U2$ $L'(i)$ $U2$ $F2$ $L'(i)$ $F2$ $R(i)$ $U2$ $R'(i)$ $U2$ $L(i)2$) exchanges the position of edge cubies $UB(i)$ and $UF(i)$ [6], while not affecting other edge cubies and all facelets. Do *Try-out* until all the edge cubies of the same kinds are grouped into solid strips.

5.3. Solving the edges

By the end of first two stages, by only twisting the surface layer, the cube can now be regarded as a 3-order Rubik's Cube and the central blocks will not be affected. There are only two clusters left: edge strips and corner cubies.

In this stage, the objective is to solve the edge strip cluster. Formula (U' L U' L' $U2$ L U' L') exchanges the position of edge strips *UR* and *UF* [6]. Formula (U' L U' L' $U2$ L U' L' F R' F' R U' R U R') exchanges the position of edge strips *UR* and *UF* and changes the orientation of them simultaneously [6]. Do *Try-out* until all the edges are solved. The way to jump out of local minima is the same as in the first stage.

5.4. Putting corner cubies to right position

First, put all the corner cubies back to their original position, regardless of their orientation; the two energy functions are suitable here. Formula (U R U' L' U R' U' L) rotates the three positions of corner cubies $URB \to ULB \to ULF \to URB$ [6]. Formula (L' U R U' L U R' U') rotates the three positions of corner cubies $URB \to ULF \to ULB \to URB$ [6]. Formula (L F' L' B L F L' $B2$ R' F R B R' F' R) exchanges two pair of cubies, *ULB* and *ULF*, *URB* and *URF* at the same time [6]. Do the *Try-out* and use the same method to jump out of local minima in the earlier stage until all eight corner cubies are set back to their original positions.

5.5. Making the orientation of corner cubies right

The last stage is to make the orientation of corner cubies right. In the solution of 3-order Rubik's Cube, formula (U R U' L' U R' U' L B L' D' L U L' D L U' B') changes the orientation of corner cubies *ULF* and *ULB* without affecting other parts [6]. Do the *Try-out* until all the corner cubies are back to their original orientation.

By now the whole Rubik's Cube is solved; the design of the process is not related to the order of the cube, which makes it extendable to any high orders.

6. Analysis of algorithm

6.1. Neighborhood search

As mentioned above, it is expected that the optimal trial step should affect the fewest cubies and match with the energy function. The ideal trial step, obviously, is to change only one element in the system, as in an Ising model. Yet due to the cube's mechanical feature, the best trial step is to exchange the positions of two adjacent cubies.

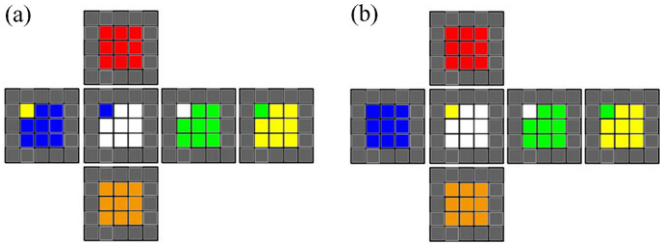

Fig. 4. Possible local minimum in the stage of solving six central blocks.

In the stage of solving six central blocks, one of the three cubies that the formula affects stays on its previous face. The energy change will be determined by the other two cubies. This means that the trial step is perfectly matched with the energy function. Thus the formula used in this stage is optimal.

The same discussion can be applied to the stage of solving the edge strips. As for the extra orientation problem, there are two randomly used formulas that cause different orientation changes for the involved edge strips; the formulas then still match perfectly with the energy function. Besides, the completeness is guaranteed by their different effects on orientation.

In the stage of solving the corner cubies, corner cubies are left to be the only cluster that can be changed. An ideal formula which exchanges only two cubies does not exist. This is a theorem proved by group theory. Therefore, the formula adopted for rotating three corner cubies should be the best one. Besides, these formulas are complete.

6.2. Local minima

By dividing the process into partial stages and clusters, and using proper formulas as trial steps, the number of local minima has been reduced greatly. Yet, there are still several 'long period' traps, one of them is shown in Fig. 4(a).

To avoid trapping the Metropolis' importance sampling allows accepting the trial steps of energy rise according to ratio of Boltzmann probability distributions between new and old states. Furthermore, simulated annealing scheme elevates such acceptance probability by increasing the temperature parameter in Boltzmann probability distribution firstly and followed by cooling down temperature.

But, for Rubik's Cube the direct application of the conventional simulated annealing scheme would damage the restored parts and the cube would be highly possible to drop into several adjacent local minima repeatedly. Therefore, we abandon to use Boltzmann distribution and temperature parameter; instead we design another effective cooling procedure.

As mentioned before, we consider two energy expressions, E_1 and E_2. E_2 makes a cubie move to the original state much faster since the energy increases greatly with distance, yet the number of possible local minima increases. For example, in the stage of solving six central blocks, one possible situation is shown in Fig. 4(a). If E_2 is adopted, all possible trial steps will result in an energy increase. This deadlock can be broken by using E_1, because by E_1, the change of energy value from the farthest position to the second nearest position equals to that from the second nearest position to the nearest position. So, when trail steps that maintain energy are accepted, there will be more paths leaving the old state, such as exchange two cubies that haven't been restored in Fig. 4(b). However, adopting E_1 may end up in the situation that two cubies in their farthest position are left to be exchanged while others have already been restored. In conclusion, an efficient method is to begin with the quadric energy model. When trapping happens,

we can switch to linear energy model. After a while, change back to E_2. Similar cooling schedules are used in other stages.

6.3. A special case in SA

In the first and third phase of the algorithm mentioned above, there are formulas that exchange the position of adjacent corner/edge cubies. And in the fifth phase, the formula changes the orientation of two adjacent corner cubies. In such cases, formulas that involve any two cubies in a cluster can be found. For example, to exchange two cubies, cubie A and cubie B which are not adjacent to each other, we only need to find a path composed of several adjacent cubies and connect A and B, for example A–X–Y–B. Exchange A and X, A and Y, A and B, then Y and B, X and B. The effect of the five steps exchanges the position of A and B, while X, Y have been restored after being moved. Thus, we have the formulas that change any of the two cubies in one cluster. When this goal is achieved, the 24 cubies can move randomly to any of the 24 positions without other limitation, the conception of distance and energy therefore loses its meaning. The algorithm then can be simplified as follows: for one cubie A that hasn't been restored, exchange it with the cubie B that occupies the original position of cubie A.

6.4. Parallel algorithm

Because of nearly quadric behavior of time with order as will be shown later, most of time is then certainly spent on solving central blocks. As discussed above, the formula used in this stage only affects three face cubies that are within the same cluster. In other words, clusters are independent of each other. This makes it possible to complete this stage in any sequence of the clusters and is suitable for a parallel computation.

There are two concerns for a parallel programming, i.e. data transmission efficiency and computing resource required by each process [7]. Our program is of master–slave parallel structure. A slave process deals with one cluster at a time and transmits the sequence of moves back to master process to operate. If the state of the whole cube would be transmitted to every slave process and every slave process would handle the whole cube, a lot of time and resources would be required, since we consider the order as high as 5001 here.

We have found an excellent solution to this difficulty, that is, a low-order cube can be used to replace the higher-order cube in parallel implementation: The formula $(R(j)\ U\ L'(i)\ U'\ R'(j)\ U\ L(i)\ U')$ is only related to the value of i and j, and regardless of the order, n, of a cube. So the positions of a specific cluster in a relatively low-order cube can be chosen to represent the position for other clusters in other higher-order cubes. For example, use position $(2,5)(3,2)(6,3)(5,6)$ in each face of 7-order cube to represent position $(i,n+1-j)(j,i)(n+1-i,j)(n+1-j,n+1-i)$ on each face of higher-order cubes. Every slave process receives data of one cluster, that is, 24 cubies, and put them in corresponding positions in a 7-order Rubik's Cube. The slave process will solve this cluster using the formula in which $i=2$ and $j=3$ and every move of the 7-order cube will be recorded. When one substituted cluster is solved, all moves are transmitted back to the master process. The master process operates the extra-high-order cube accordingly, using the formula with original i and j of the specific cluster. The key for this operation is the relationship between the positions in the high-order cube and the 7-order cube. In this way, each process only handles a 7-order cube, and the transmission of data is simplified.

7. Computational results

Figs. 5(a) and 5(b) show computing time for solving process versus order of cube performed with a laptop computer and a par-

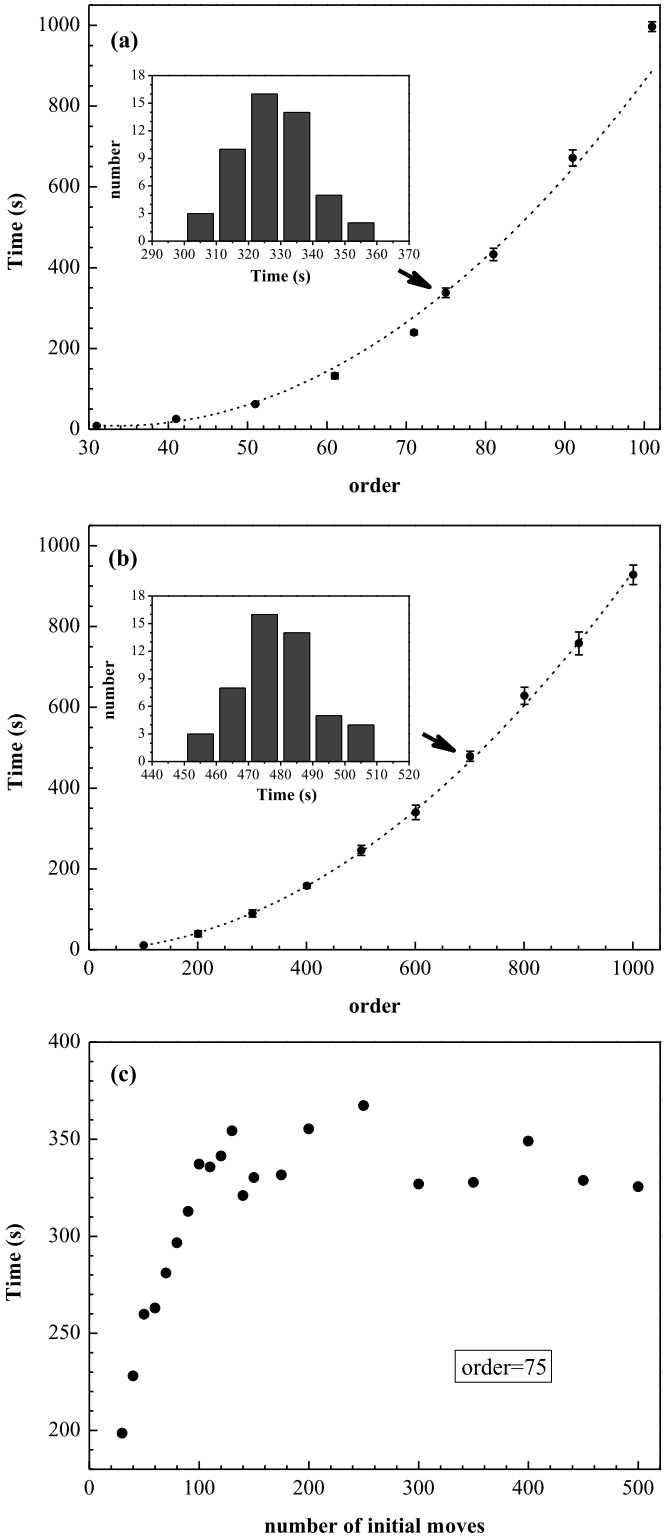

Fig. 5. Computing time versus order of cube performed with: (a) a laptop computer; (b) a parallel cluster with 32 nodes. Dashed lines are polynomial fits up to quadric term. The arrow indicated data represent 50 times of calculation, whose computing time distribution is shown by the inset figure, for the same order but with different thermalized initial states, while other data are for 5 times of repetition with different initial states. (c) The computing time for solving a cube that is partially disordered versus the number of initial moves to disorder the cube from original state.

allel cluster. The thermalization of initial state has been prepared by random rotating the Cube from original state so that the potential energy becomes relatively steady with rotations. The time in preparation of initial state has been excluded from this calculation time. Clearly the time is quadric in order, which is much slower than the increasing rate of number of microstates A with order. The calculation time for solving some low-order cubes from 31 to 101 by a PC (Lenovo Z460 laptop) is shown in Fig. 5(a). For each order we have repeated calculation by 5 times with different thermalized initial states, except for the order of 75 the repetition number is 50. The computing time consists of three parts: solving the central blocks, grouping edges, and solving the quasi 3-order cube. The number of clusters that are composed of face facelets is proportional to n^2, and the number of clusters that are composed of edge cubies is proportional to n. Fig. 5(b) is for solving some high-order cubes from 101 to 1001 by using 31 slave processes on a parallel cluster. For each order we have repeated calculation by 5 times with different thermalized initial states, except for the order of 701 the repetition number is 50. Some extra-high-order cubes are also solved but in a single calculation: A 5001-order cube takes 1877 s with 8 nodes. All the parallel calculation data presented here have been performed with a supercomputer ("Magic Cube", Dawning 5000A) at Shanghai Supercomputer Center and other individual calculations done with a parallel computer at Supercomputing Center of USTC derive the similar curve as in Fig. 5(b). In above calculations a fully disordered initial state is randomly chosen at a sufficient large number of moves from original state. Fig. 5(c) indicates that if the number of moves for preparation of initial state is small so that the chosen initial state is only partially disordered, the calculation time for solving the cube can be even short as expected.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 11074232 and 10874160), "973" projects (Nos. 2011CB932801 and 2012CB933702) and "111" project. We thank Supercomputing Center of USTC and Shanghai Supercomputer Center for their support to parallel computations, and Dr. H.M. Li for his technical support.

References

[1] N. El-Sourani, Design and benchmark of different evolutionary approaches to solve the Rubik's Cube as a discrete optimization problem, Diploma thesis, WWU Muenster, Germany, 2009.
[2] D. Joyner, Adventures in Group Theory: Rubik's Cube, Merlin's Machine & Other Mathematical Toys, 2nd ed., Johns Hopkins Univ. Press, 2008;
G. Betsch, The Mathematical Intelligencer 27 (2) (2005) 92, doi:10.1007/BF02985810.
[3] Number of combinations to the Rubik's Cube and variations, http://www.speedcubing.com/chris/cubecombos.html.
[4] R. Korf, Finding optimal solutions to Rubik's cube using pattern databases, in: Proc. Workshop on Computer Games (W31) at IJCAI-97, Nagoya, Japan, 1997, pp. 21–26.
[5] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, Science 220 (1983) 671–680.
[6] Chris Hardwick's Rubik's Cube page, http://www.speedcubing.com/chris/index.html.
[7] B.P. Lester, The Art of Parallel Programming, 2nd ed., 1st World Publishing Inc., 2008.