Journal of Mathematical Research and Exposition
Vol. 10, No.4, Nov., 1990

An Algorithm for Minimal Number-Grouped Partitions of Random Permutations*

Ding Kenquan

(Dept. Math., Univ. Wisconsin-Madison, Madison, WI 53706)

Abstract

In this paper, by means of a kind of strongly isomorphic cutting-branch technique, an algorithm for the problem of finding the minimal number-grouped partitions of random permutations is given, and the open problem in [1] is solved.

I. Introduction

The studies on the problem of minimal number-grouped partitions of random permutations started around the center of 1960's. In 1978, it was announced by a group of mathematicians in Academia Sinica as an open problem in [1]. For some special cases of this problem, recently, Yongjin Zhu and Ruopeng Zhu obtained an efficient algorithm under the condition of quasi-orderedness [2], and Kequan Ding solved the problem with the pseudo-ordered condition [3]. In particular, the study on the basic subroutine, the scheme-moving-down algorithm, shows that a generalization of this algorithm can be applied to solve the problem of minimal conforming partitions of any labelled finite posets (see[4]). On the other hand, the original problem of minimal number-grouped partitions of random permutations seems very difficult. In 1986, on the China-USA International Conference of Graph Theory (Jinan), Guozhi Xu, Qinghua Chen and Jiyong Liu proposed a conjecture that this problem is *NP*-complete. Because of this, they studied the approximation solutions of this problem. For the accurate solution problem, Jiyong Liu gave an algorithm based on the dynamic programming method, before. But, the computational complexity of that algorithm is extremely high, which is the factorial of the length of sequences. Hence, Qinghua Chen pointed out that, on the conference, this algorithm is not useful, practically.

In this paper, we study the accurate solution problem. By means of introducing a kind of strongly isomorphic cutting-branch technique, we get an algorithm

* Received June 24, 1988.

for the problem with time complexity $O(mn2^n)$, where n is the number of different digits in the sequences. According to the engineering background of our problem, n is usually less than or equal to 10 and the length m of a sequence is less than or equal to 70. Therefore, the algorithm given here is applicable, and the open problem in [1] is solved.

Without special statement, we shall use the definitions and symbols as in [2,3] in our discussion.

2. N-Generating Tree

In this section, the concept of n-generating tree is introduced and its properties are studied, which will be used in the rest part of our discussion.

Definition 2.1 (n-generating tree) Let T_n be a rooted tree of height n. If it satisfies the following condition that

$$|\Gamma(v)| = n - r(v), \quad \forall v \in V(T_n),$$

then T_n is called an n-generating tree. Usually, we denote its root as v_0.

By definition, it is easy to see that the n-generating tree has the following properties.

i) $|V_i(T_n)| = \binom{n}{i} i!$

ii) Let $T_n(v)$ be a subtree of T_n, whose root is v, then $T_n(v)$ is an $(n-r(v))$-generating tree.

Definition 2.2 ((m, n) permutation) Let $\pi: a_1, a_2, \cdots, a_m$ be a repeatable permutation which is arranged at random with n numbers, say, $1,2,\cdots,n$. Without loss of generality, the discussion is confined to such a case that each number in the set $\{1,2,\cdots,n\}$ must appear in π at least once. Denote $\pi(i) := a_i$, $1 \leqslant i \leqslant m$. Suppose that there are two subsequences of π, say, π' and π'', which satisfy that for any given natural number k $(1 \leqslant k \leqslant n)$, the frequency of k appeared in π' is equal to that in π''; then π' is called to have the same content as that of π''.

Theorem 2.3 Let T_n be an n-generating tree with its root v_0. There exists a labelling function φ defined on $V(T_n)$ such that the set of all labelling sequences on the n-paths of T_n coincides with the set $S(n)$ of all (n, n) random permutations.

Proof Let $S_i(n) := \{\pi \in S(n) \mid \pi(1) = i\}$, $1 \leqslant i \leqslant n$. Clearly, the family $\{S_i(n)\}$, $1 \leqslant i \leqslant n$. forms a partition of $S(n)$, i.e.,

$$S_i(n) \cap S_j(n) = \varnothing, \quad i \neq j;$$

and

$$\bigcup_{i=1}^{n} S_i(n) = S(n).$$

Thus, by the property ii) of T_n and the principle of mathematical induction, the proof is completed.

Definition 2.4 Let (T, φ) be a labelled n-generating tree, where φ is a labelling function defined on $V(T)$. Let $v_1, v_2 \in V(T)$, and $T(v_1), T(v_2)$ be two subtrees of T, with their roots v_1 and v_2, respectively. If there exists a bijection ζ from $V(T(v_1))$ to $V(T(v_2))$, such that

i) $\forall v', v'' \in V(T(v))$, the necessary and sufficient condition that $(v', v'') \in E(T(v_1))$ is $(\zeta(v'), \zeta(v'')) \in E(T(v_2))$.

ii) $\varphi(v') = \varphi(\zeta(v'))$ holds true for any $v' \in V(T(v_1)) \setminus \{v_1\}$.

then $T(v_1)$ is said to be strongly isomorphic to $T(v_2)$, and is denoted as $T(v_1) \cong T(v_2))$.

It is evident that the strongly isomorphic relation is an equivalent relation on the set of subtrees of (T, φ). By the definition, it is easy to show the following simple, but very useful facts.

Lemma 2.5 Let T_n be an n-generating tree. For any $v_1, v_2 \in V(T_n)$ the necessay and sufficient condition for $T_n(v_1) \cong T_n(v_2)$ is the labelling sequence on Pv_1 has the same content as that of Pv_2.

Corollary 2.6 Let T_n be the same as above. For any $v \in V(T_n)$, there are $(r(v)!)$ subtrees on T_n that are strongly isomorphic to $T_n(v)$.

Lemma 2.7 Let T_n be the same as above. For any $v, v' \in V(T_n)$, there are $(r(v') - r(v))!$ subtrees on T_n that are strongly isomorphic to $T(v')$, with the convention that

$$(-a)! = 0, \text{ for } a > 0.$$

Lemma 2.8 Let T_n be the same as above. For any given natural number h, classify the members in $V(T_n)$, the necessary and sufficient condition for that v' and v'' are in the same class is $T_n(v') \cong T_n(v'')$. Denote the family of all these classes as $\{V_{hi}\}_{i=1}^{l}$, then

$$L = \binom{n}{h}.$$

3. The Strongly Isomorphic Cutting-Branch Strategy and The Algorithm

By the optimal property theorem of the scheme-moving-down algorithm in [2], so far as a random permutation is concerned, if $\{S_i\}_{i=1}^{t}$ is a minimal number-grouped partition of π, denote

$$\pi' := S_1 \circ S_2 \circ \cdots \circ S_t,$$

then $t := |P_{\pi'}(\pi)|$, where $P_{\pi'}(\pi)$ is the scheme-moving-down partition of π according to π'. Hence, the problem of finding a minimal number-grouped partition of a random permutation can be divided into two speps.

First, find the scheme-moving-down partition of π according to each member

in $G(\pi)$. Then, among all these partitions, find one of them which is formed with fewest parts. It is clear that such a method is not valuable for its time complexity is too high to use, $O(m^2(n!))$. But, the method gives us a heuristic clue for formulating our algorithm. Using the strongly isomorphic cutting-branch strategy, we are successful in making an implicit enumeration which is fit for the purpose of reducing the time complexity. The key point of this strategy is to make comparisons among the paths ended at points which are of the same rank and their adjacent subtrees are strongly isomorphic with each other.

From the discussion in section 2, one can see that two paths on T_n, say, Pv_1, and Pv_2, are comparable, iff

$$T_n(v_1) \cong T_n(v_2).$$

Thus, the minimal number-grouped partitions of random permutations can be obtained with the following algorithm (without running the risk of causing confusion, we identify a number-group sequence and its condensed sequence, and do not distinguish a path Pv in T_n with its labelling sequence).

Algorithm G

Let π be an (m, n) random permutation, and T_n be an n-generating tree.

1) $i \leftarrow 1$, $j \leftarrow 1$, $T \leftarrow T_n$;

2) $i = n+1$? If so, stop. Otherwise, $j \leftarrow 1$;

3) $V_i := V_i(T) \setminus \{v_{ik}\}_{1 \le k \le j-1}$;

4) $V_i = \varnothing$? If so, $i \leftarrow i+1$, $j \leftarrow 1$, turn to 2);

5) $\forall v \in V_i$, $v_j \leftarrow v$; let

$$C_{ij} := \{v' \in V_i(T) \mid T(v') \cong T(v_j)\};$$

6) find $v_{ij} \in C_{ij}$ such that $Pv_{ij} > Pv'$, $\forall v' \in C_{ij}$;

7) cut off the adjacent branch of v', for any $v' \in C_{ij} \setminus \{v_{ij}\}$, and get a new tree T';

8) $T \leftarrow T'$, $j \leftarrow j+1$, turn to 2);

4. The Basic Lemma

In this section, we prove the following basic lemma which was first announced, without proof, in [5].

Lemma 4.1 Let π be an arbitrary (m, n) random permutation, then a minimal number-grouped partition of π can be found with **Algorithm** G. The time complexity of the algorithm is $O(m^2 n 2^{n-2})$.

Let $V_i^*(T)$ be a subset of $V_i(T)$, which is formed with the points left after the performance of the algorithm in $V_i(T)$. Then, no pair of potnts in $V_i^*(T)$ have their adjacent subtrees strongly isomorphic to each other. Meanwhile, any subtree of T_n must be strongly isomorphic to one of the subtrees whose root lies in $V_i^*(T)$. Note that any pair of subtrees of T_n whose root lie in $V_n(T_n)$ must be

strongly isomorphic to each other. Hence, the performance of the algorithm leads to a single n-path of T_n. By the comparison-in-part criteria in [3], we know that thc scheme-moving-down partition of π according to the n-path left is a minimal number-grouped partition of π. Furthermore, the above discussion and the conclusion of Lemma 2.8 imply that after the cutting-branch processes in all members of $\{V_i(T)\}_{1<i<k-1}$, there are

$$C_{k-1}^{n}=\binom{n}{k-1}.$$

points left in $V_{k-1}(T)$, exactly. Note that for each pair of these points, their adjacent subtrees are not strongly isomorpnic. Then, they have $(n+k+1)C_{k-1}^{n}$ sons in $V_k(T)$, totally. Classify these son-points according to the contents of their corresponding k-paths, i.e., all those whose corresponding k-paths have the same content are in a same class, otherwise, they belong to different classes.It is easy to see that each of the classes has k members. Thus, the cutting-branch operation in $V_k(T)$ can be realized by $(k-1)$ times of comparisons in each of the class. On the set $V_k(T)$, we need to compare

$$(((k-1)(n-k+1))/k)C_{k-1}^{n}$$

times. Through simple combinatorial calculation, we find that the performance of the Algorithm G can be fulfilled with

$$\sum_{k=1}^{n}\frac{(k-1)(n-k+1)}{k}C_{k-1}^{n}=(n-2)2^{n-1}+1$$

times of comparisons among paths. As each comparison can be realized by the scheme-moving-down algorithm which can be carried out in $m(m+1)/2$ times of comparisons, hence, the time complexity of Algorithm G is $O(m^2n2^{n-2})$, and the proof is completed.

5. Decomposibility of Li's Algorithm

It is easy to see that Li's algorithm is the basic subroutine of our optimization process. Generally, the computation time of the algorithm is $O(m^2)$.Because of this, as a subroutine, it is rather slow. But, the studies on the structure of the algorithm show that it is decomposible. And this property enables us to reduce the computational time of **Algorithm** G.

Theorem 5.1 (decomposibility of Li's algorithm) Let π and π' be (m, n) random permutations with the same content, $\pi 1=\pi'(1)\pi'(2)\cdots\pi'(m-1)$.Denote the term of π corresponding to $\pi'(j)$, in the process of execution of Li's algorithm on π according to π', as $\pi(i_j)$, $1\leqslant i\leqslant m$. Let $\{\eta_j\}_{1<j<k}$ be the scheme-moving-down partition of π according to $\pi 1$. Then, the last term of η_k is $\pi(i_{m-1})$ and the scheme-moving-down partition of π according to π' is as follows:

$$\{\eta_j\}_{1<j<k-1}\cup\{\eta_k\circ\pi(i_m)\},\quad \text{if } i_{m-1}<i_m,$$

$$\{\eta_j\}_{1<j<k}\cup\{\pi(i_m)\}\},\ \text{if}\ i_{m-1}>i_m.$$

Proof It follows from the definition of Li's algorithm, directly.

Let π be a given random permutation, π' and π'' two random permutations of length k and

$$\pi 1=\pi'(1)\pi'(2)\cdots\pi'(k-1)=\pi''(1)\pi''(2)\cdots\pi''(k-1).$$

In order to compare the scheme-moving-down partitions of π according to π' and π'', respectively, by the theorem above, we need only to execute the following algorithm, in stead of Li's.

Algorithm G

a) find the scheme-moving-down partition of π according to $\pi 1$, say $\{\eta_j\}_{1<j<h}$

b) according to the rule of Li's algorithm, find the corresponding term $\pi(i_k)$ of $\pi'(k)$ in π.

c) according to the rule of Li's algorithm, find the corresponding term $\pi(j_k)$ of $\pi''(k)$ in π.

d) by the methed mentioned in Theorem 5.1, find the scheme-moving-down partition of π according to π' and π'' from the partition $\{\eta_j\}_{1<j<h}$. And, we define that $\pi'<\pi''$ iff the number of parts in the scheme-moving-down partition of π according to π' is less than that according to π'', or these two numbers are equal and $i_k<j_k$.

Obviously, the execution of b) and c) needs $2(m-k+1)$ times of comparisons; i.e., $O(m)$ times of comparisons, and the execution of d) needs $O(1)$ times of comparisons.

For example, $\pi=3213221$, $\pi'=11223$, $\pi''=11222$. Thus, the scheme-moving-down partitions of π according to $\pi 1$, π' and π'', respectively, are as follows:

1	1	1	1	1	1	
2	2	2	2	2	2	2
		3				

Clearly, we have $\pi''<\pi'$.

Note that in the execution of **Algorithm** G, whenever we compare two branches of the tree, the only difference between them may occur at the last position. Hence, we can use the scheme-moving-down partition of π according to the common part of the two branches, which is obtained in the previous loop of **Algorithm** G, and then, execute b), c) and d) of **Algorithm** C. In this way, a comparison between two branches needs only $O(m)$ times comparisons of keys. This proves the main result of the paper.

Theorem 5.2 A minimal number-grouped partition of any (m, n) random permutation can be obtained by **Algorithm** G, with its time complexity $O(mn2^n)$.

Similar arguments lead to an improved form of the main result in [3].

Theorem 5.3 Let π be any (n, n) random permutation. A minimal pseudo-ordered partition of π can be obtained by an effcient algorithm, with its time complexity $O(n^3)$.

6. Conclusion

In the theoretical opoinion, **Algorithm** G is exponential. Thus, when n increases, the computation time of the algorithm will increase, rapidly. But, for the cars marshalling practice, in the railway system of China, generally, what we need to do is to find the minimal number-grouped partitions of random permutations for the case that $n \leqslant 10$. Therefore, the algorithm seems applicable in practice. Again, note that in the process of the algorithm, the only factor which has some thing to do with the number m is the subroutine of the decomposed scheme-moving-down algorithm, **Algorithm** C. This shows that the time complexity of the algorithm is stable with respect to m, for any given n.

In addtion, in the practical computation, it is not necessary to keep all the previous scheme-moving-down partitions in the memory unit of a computer. Here, we need only to keep their numbers of parts and the address of the last term of their last part in π. After executing the **Algorithm** G, we get the optimal standard sequence. Then, we can find the scheme-moving-down partition of π according to this standard sequence, by Li's algorithm. Therefore, the reduction of time complexity, by the decomposibility of Li's algorithm, is essential which does not cost more space resources than before.

References

[1] The 2nd Group of Operations Research, Institute of Mathematics, Academia Sinica, The first research on the mathematical methods in the problem of cars marshalling, Acta Mathematicae Applicatae Sinica, 1:2 (1978), 91—105.

[2] Zhu Yongjin, Zhu Ruopeng, Sequence reconstruction under some order-type constraints, Scientia Sinica A, 2 (1983), 119—125.

[3] Ding Kequan, The minimal pseudo-ordered partitions of random permutations, Chinese Jounal of Operations Research, 1 (1985), 63—64.

[4] Ding Kequan, A generalized scheme-moving-down algorithm and a dual scheme-moving-down algorithm, Journal on Numerical Methods and Computer Applications, 8:2 (1987), 115—121.

[5] Ding Kequan, An algorithm for the problem of minimal number-grouped partitions of random permutations (Research Announcement), Advances in Mathematics (Beijing), 15:1 (1986), 125—126.

随机排列最优成组剖分的算法

丁 克 诠

（美国威斯康星一 麦迪逊大学数学系）

摘 要

本文研究随机排列的最优成组剖分问题．这一问题源于铁路列车的最优调度计划方法的设计问题．寻找切实可行的有效算法是问题的焦点．1978年这一问题被列入文献[1]的公开问题之一．1986年许国志、陈庆华和刘继勇提出猜测：此乃NP-完全问题，即多项式时间的算法可能不会存在，除非NP=P．

本文引入一种强同构剪枝策略，以标号树形上的隐式枚举法为工具，得到了上述问题精确最优解的一个算法．其计算时间复杂度为$O(n^3 2^{n-2})$，其中n为随机排列中相异数字的个数．算法在给定n的条件下，关于随机排列的长度是二次稳定的．因此，由铁路工程实际需要来看，该算法是切实可行的．至此，文献[1]的公开问题得解．特别地，本文将此类问题归入两步优化结构模式，使问题的几何特征有明显的表露．这一策略在相关的领域中有着良好的应用前景．

接 515 页

Some Finite-Dimensional Involutive Systems with Polynomial Forms

Ma Wenxiu

(Institute of Mathematics, Fudan Universiy)

Abstract

In this paper, starting from the combination of two involutive systems, we consider separately some systems of polynomial functions: $\{I_m^{(1)} = B_m + R_m\}_{m=0}^{\infty}$, $\{I_m^{(2)} = B_m + S_m\}_{m=0}^{\infty}$, $\{I_m^{(3)} = B_m + T_m\}_{m=0}^{\infty}$ and so on; and analyse carefully the sufficient and necessary conditions of the involution of three systems of functions $\{I_m^{(i)}\}_{m=0}^{\infty}$ $(1 \leq i \leq 3)$ with general coefficients. Furthermore, we present concrete forms of the involutive systems hidden in $\{I_m^{(i)}\}_{m=0}^{\infty}$ $(1 \leq i \leq 3)$; and thus obtain six kinds of nontrivial involutive systems of functions, which include a few involutive systems discussed in the literature. Based upon these involutive systems, we can generate a lot of new finite-dimensional Hamiltonian systems which are completely integrable in the sense of Liouville.