© International Journal of Combinatorial Optimization Problems and Informatics, Vol. 8, No. 1, Jan-April 2017, pp. 19-30. ISSN: 2007-1558.

# Enumeration and Generation of Permutations with a Partially Fixed Order of Elements

Yuri G. Stoyan[1], Igor V. Grebennik[2], Viacheslav V. Kalashnikov[3,4,5*], and Oleksandr S. Lytvynenko[2].

[1]*A.N. Podgorny Institute for Mechanical Engineering Problems, Kharkiv, Ukraine 61046*
[2]*Kharkiv National University of Radio Electronics, Kharkiv, Ukraine 61166*
[3]*Tecnológico de Monterrey (ITESM), Campus Monterrey, Monterrey, Mexico*
[4]*Central Economics and Mathematics Institute (CEMI), Moscow, Russia 117418*
[5]*Sumy State University, Sumy, Ukraine 40007*

*stoyan@ipmach.kharkov.ua, igorgrebennik@gmail.com, kalash@itesm.mx, litvinenko1706@gmail.com*

**Abstract.** A specific class of permutations – permutations with partially fixed order of elements – has been described. A procedure of enumeration and generation of this class has been developed. Steps of the algorithm have been established to be well-defined, and its complexity has been evaluated. Some important particular classes of permutations, such as alternating and unimodal ones are special cases of the permutations described. This algorithm can be used for generating other combinatorial sets, e.g. arrangements and combinations.

**Keywords:** Enumeration and Generation of Permutations, Partially Fixed Order of Elements.

## 1. Introduction

Numerous monographs and papers on the exhaustive generation of combinatorial sets (*e.g.*, [1]–[4]) mainly deal with classical combinatorial objects such as permutations, trees etc. However, in many real-world problems, generation of non-classical combinatorial objects or classical ones with additional special properties is needed (*cf.*, [3]).

Permutation sets are considered the most common and well-examined combinatorial sets. Unrestricted permutations generation algorithms have been developed in numerous books and papers, *cf.*, [1]–[2]. In many cases, generation problems are solved for sets of permutations with special properties, like sets of permutations with a given number of cycles, Bell permutations, or alternating permutations [4]–[5]. Problems of enumeration of permutations with a given order of elements have been explored since the early 1970s. In some papers, problems of enumeration for some classes of permutations with prescribed up-down structure are solved. These permutations are called permutations with a prescribed pattern; they are special cases of permutations with a fixed order of elements. Enumeration problems for permutations having a prescribed sequence of rises and falls are considered in [6] and many others. A permutation with a signature $Q=(q_1,q_2,...q_{n-1})$ where $q_i$ is either 1 or – 1, is a permutation $P=\pi_1,\pi_2,...\pi_n$ of integers 1 to *n*, so that $\pi_i<\pi_{i+1}$ if $q_i=+1$, $\pi_i>\pi_{i+1}$ otherwise, for all $i=1,2,...,n-1$. Alternating permutations [4]–[5] are an example of permutations with the signature (1, –1, 1, –1, ...). In [6], a problem of generation of permutations with a given signature is solved.

In this paper, we are going to consider a generalization of permutations with a given signature – permutations with partially fixed order, where "greater-less" relations are determined not for all pairs of neighboring elements but only for some of them.

* This research was partially financed by the SEP-CONACYT grant CB-2013-01-221676, Mexico

Received Jun 19, 2016 / Accepted Dec 10, 2016

The rest of the paper is organized as follows. Permutations with a partially fixed order of elements are defined and studied in Section 2. Section 3 deals with the formal description of the proposed algorithm for generating the whole list of permutations with a partially fixed order of elements. A particular case of such permutations is considered in Section 4. Concluding remarks and the list of references are in Sections 5 and 6, respectively. Finally, the detailed presentation of the steps of the developed algorithm and some semantic features of the latter can be found by the interested reader in Appendix *A* (Section 7), while a link to a paper assessing and discussing the algorithm complexity is provided in Appendix *B* (Section 8).

## 2. Permutations with a Partially Fixed Order of Elements

Consider a set of generating elements $A=\{a_1,a_2,...,a_n\}$, $a_1<a_2<...<a_n$, $a_i\in R^1$, $i\in J_n$, $J_n=\{1,2,...,n\}$, and let $P_n$ denote a set of all possible permutations of entries $a_1,a_2,...,a_n$. For an arbitrary permutation $\pi=(\pi_1,\pi_2,...,\pi_n)\in P_n$, the following relationships are valid:

$$\pi_1\,\rho_1\,\pi_2\,\rho_2\,...\rho_{n-1}\,\pi_n,\ \rho_i\in\{<,>\},\ i\in J_{n-1}. \quad \textbf{(1)}$$

In other words, Eq. (1) means that, for each $i\in J_{n-1}$, exactly one of the inequalities holds:

$$\pi_i<\pi_{i+1}, \quad \textbf{(2)}$$

or

$$\pi_i>\pi_{i+1}. \quad \textbf{(3)}$$

**Definition 1**. The sequence $\rho(\pi)=(\rho_1,\rho_2,...,\rho_{n-1})$ is called the *order of elements* in the permutation $\pi\in P_n$. A permutation with an order of fixed elements $\pi^i$ is called *permutation with a fixed order*. Permutations with a fixed order are identical to the permutations with a given signature described by Roeiants van Baronaigien and Ruskey [6]. For example, permutations with the order $\rho(\pi)=(<,>,>,<)$ are identical to those with signature $Q=(-1,+1,+1,-1)$.

**Definition 2**. An arbitrary subsequence $\pi^i$, i.e. a sequence $r(\pi)=(\rho_{i_1},\rho_{i_2},...,\rho_{i_k})$, $i_1<i_2<...<i_k$, $k\in J_{n-1}$, is called a *partially fixed* (or *partially set*) order of elements for permutation $\pi\in P_n$.

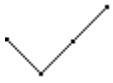
A complete order of elements $\pi^i$ determines relationships between $\pi_i$ and $\pi_{i+1}$ for all positions $i\in J_{n-1}$, while the partially fixed order $r(\pi)$ determines it only for positions $i\in\{i_1,i_2,...,i_k\}$. The order of elements of the permutation $\pi\in P_n$ can be formalized in various ways. Roeiants van Baronaigien and Ruskey [6] made such formalization via signatures. We will describe an order of elements in permutations using a *set of descent* introduced by Stanley [5].

**Definition 3.** The *set of descent* $D(\pi)$ of permutation $\pi\in P_n$ is determined as in [5] $D(\pi)=\{i\,|\,\pi_i>\pi_{i+1}\}$, $i\in J_{n-1}$. The complementary set $\bar{D}(\pi)=J_{n-1}\setminus D(\pi)$ is denoted with $F^i$, $i\in J_{n-1}$.

In this paper, a special subset of $D(\pi)$ comprising positions $i\in\{i_1,i_2,...,i_k\}$ is used. Let us denote it by $D'(\pi)=\{i\in\{i_1,i_2,\ldots,i_k\}\,|\,\pi_i>\pi_{i+1}\}\subseteq D(\pi)$. Next, $\bar{D}'(\pi)=\{i_1,i_2,...,i_k\}\setminus D'(\pi)$ can be rewritten as follows: $\bar{D}'(\pi)=\{i\in\{i_1,i_2,\ldots,i_k\}\,|\,\pi_i<\pi_{i+1}\}\subseteq\bar{D}(\pi)$. Subsets $D'(\pi)$ and $\bar{D}'(\pi)$ explicitly determine the partially fixed order of elements for each permutation $\pi\in P_n$. The partially fixed order of elements $r(\pi)$, on the one side, and the subsets $D'(\pi)$ and $\bar{D}'(\pi)$, on the other side, are explicit functions of each other. For example, if $r(\pi)=(\rho_2,\rho_4,\rho_5)=(>,<,>)$, then $D'(\pi)=\{2,5\}$ and $\bar{D}'(\pi)=\{4\}$. This statement is also true for the

complete order of elements $\rho(\pi)$ as a special case of the partially fixed order $r(\pi)$. A permutation satisfies the partially fixed order $r(\pi)$ in case when for all positions $i \in \{i_1, i_2, ..., i_k\}$ the difference $\pi_i - \pi_{i+1}$ fits the relation $r(\pi)$. However, the relation $\pi_i - \pi_{i+1}$ is not defined for $i \notin \{i_1, i_2, ..., i_k\}$.

**Example 1**. The partially fixed order $r(\pi)$ that corresponds to the sets $D'(\pi) = \{1\}$ and $\bar{D}'(\pi) = \{2\}$ has been defined on a set of generating elements $A = \{1,2,3,4\}$ for $k = 2$ positions $i \in \{1,2\}$. For position $i = 3$ both $\pi_2 < \pi_3$ and $\pi_2 > \pi_3$ are acceptable. Thus, the partially fixed order $r(\pi)$ corresponds to $2^{n-k-1} = 2^{4-2-1} = 2$ complete orders $\rho(\pi)$ that can be presented as follows:

 $D(\pi) = \{1\}$, $\bar{D}(\pi) = \{2,3\}$ and  $D(\pi) = \{1,3\}$, $\bar{D}(\pi) = \{2\}$.

In both cases, for $i \in \{1,2\}$, the relation $\pi_i - \pi_{i+1}$ satisfies the assigned one; the set of descents includes $D'(\pi)$, and $\bar{D}(\pi)$ comprises $\bar{D}'(\pi)$.

Now let us consider the problem of enumeration of permutations $\pi \in P_n$ with a partially fixed order of elements $r(\pi)$. The number of permutations satisfying the given partially fixed order $r(\pi)$ is equal to the total of the number of permutations satisfying each of $2^{n-k-1}$ complete orders $\rho^j(\pi)$, $j = 1,2,...,2^{n-k-1}$. Then, we need to find all complete orders $\rho^1(\pi), \rho^2(\pi), ..., \rho^{2^{n-k-1}}(\pi)$ that correspond to the partially fixed order $r(\pi)$ to determine the number of permutations satisfying each of the complete orders and to sum them all to obtain their total. Permutations with a given complete order of elements $\rho(\pi)$ are identical to permutations with a given signature. Therefore, to calculate a number of these permutations, the previous results obtained for permutations with a given signature can be used, e.g., [7].

The number of permutations with a given complete order of elements $\rho(\pi)$ can be also calculated through its set of descent. To enumerate permutations $\pi \in P_n$ with the order of elements $\rho(\pi)$ we are going to calculate the total number $N(\rho(\pi))$ of all permutations $\pi \in P_n$ having $D(\pi) = S$. Following [5], for a fixed $S = \{s_1, s_2, ..., s_k\} \subseteq J_{n-1}$ let the number of permutations $\pi \in P_n$ having $S$ as its (exact) set of descents be denoted with $\beta_n(S)$. Stanley [5] developed a formula to determine $\beta_n(S)$:

$$\beta_n(S) = \sum_{1 \le i_1 < i_2 < ... < i_j \le k} (-1)^{k-j} \binom{n}{s_{i_1}, s_{i_2} - s_{i_1}, ..., n - s_{i_j}}, \quad (4)$$

where $\binom{n}{n_1, n_2, ..., n_k} = \frac{n!}{n_1! \cdot n_2! \cdot ... \cdot n_k!}$, $j \in J_k$. Let $\rho(\pi^0)$ be the order of elements in the permutation $\pi^0 \in P_n$, with $D(\pi^0) = S = \{s_1, s_2, ..., s_k\} \subseteq J_{n-1}$. The number $N(\rho(\pi^0))$ of all permutations $\pi \in P_n$ having the order of elements $\rho(\pi) = \rho(\pi^0)$ is equal to $\beta_n(S)$, i.e. $N(\rho(\pi^0)) = \beta_n(S)$ where $\beta_n(S)$ is determined by Eq. (4). By summing up the number of permutations with the order of elements $\rho^1(\pi), \rho^2(\pi), ..., \rho^{2^{n-k-1}}(\pi)$, the number of permutations with the partially fixed order of elements $r(\pi)$ is obtained:

$$N(r(\pi)) = \sum_{j=1}^{2^{n-k-1}} N(\rho^j(\pi)). \quad (5)$$

The next section describes an algorithm for generating all permutations $\pi \in P_n$ with a partially fixed order of elements $r(\pi)$.

## 3. Generation of Permutations with a Partially Fixed Order of Elements

Let us select a set of generating elements $A=\{a_1,a_2,...,a_n\}$ such as $a_1<a_2<...<a_n$, and a partially fixed order of elements $r(\pi)$ determining the sets $D'(\pi)$ и $\overline{D}'(\pi)$. All permutations $\pi\in P_n$ corresponding to $D'(\pi)$ и $\overline{D}'(\pi)$ must be generated in quantity $N(r(\pi))$ determined by Eq. (5). We use $\pi^i$ to denote the partial permutation composed of the first $i$ elements $\pi^i=(\pi_1,\pi_2,...,\pi_i)$, $\pi_i\in A$, $i\in J_n^0$, $J_n^0=\{0,1,2,...,n\}$. Here $\pi^0$ is an empty partial permutation, while $\pi^n=\pi\in P_n$ is the desired result of generation. Now we describe the algorithm **PartOrderedPerm** generating all permutations with the partially fixed order of elements $r(\pi)$.

At the beginning (zero level), an input of the algorithm is the empty permutation $\pi^0$. The algorithm has a recursive structure: at each recursion level, $i\in J_{n-1}^0$ it expands the current partial permutation $\pi^i=(\pi_1,\pi_2,...,\pi_i)$ of length $i$ by adding the next element with the number $i+1$ and thus obtaining a new partial permutation $\pi^{i+1}=(\pi_1,\pi_2,...,\pi_{i+1})$ of length $i+1$ at the next level. Consequently, at the level $n$, the desired permutation $\pi^n=\pi$ is obtained. The newly selected element $\pi_{i+1}\in A$ must satisfy the conditions below.

**Condition 1**. A new element $\pi_{i+1}$ is to be distinct from all previously selected elements of the partial permutation in question:

$$\pi_{i+1}\neq\pi_j,\ \forall j\in J_i. \quad (6)$$

**Condition 2**. If the partially specified order of elements $r(\pi)$ requires a descent in position $i$, then $\pi_{i+1}$ has to be smaller than $\pi_i$:

$$i\in D'(\pi)\Rightarrow\pi_{i+1}<\pi_i. \quad (7)$$

Otherwise, if $r(\pi)$ needs an ascent in position $i$, then $\pi_{i+1}$ must be greater than $\pi_i$:

$$i\in\overline{D}'(\pi)\Rightarrow\pi_{i+1}>\pi_i. \quad (8)$$

Finally, if the partially fixed order of elements $r(\pi)$ does not include position $i$, element $\pi_{i+1}$ can be either greater or smaller than $\pi_i$; i.e., there are no limitations for $\pi_{i+1}$:

$$i\notin D'(\pi), i\notin\overline{D}'(\pi)\Rightarrow\pi_i\rho_i\pi_{i+1},\rho_i\in\{<,>\}. \quad (9)$$

**Condition 3**. This condition being too lengthy has been exported to ***Appendix A***.

Now let us describe the main algorithm **PartOrderedPerm** that generates all permutations with a partial order of elements $r(\pi)$. A pseudo-code of this algorithm is shown in Figure 1. To generate all required permutations, the procedure **PartOrderedPerm** is called with input $\pi^0=\varnothing$. The input data for **PartOrderedPerm** are identical to input data for the procedure **Get_F** (*cf.*, ***Appendix A***). As mentioned above, at each level up to level $n-1$ the algorithm adds a new element to a current permutation $\pi^i$ while it returns the desired permutation $\pi^n$ at level $n$. At each recursion level $i\in J_{n-1}^0$, the following operations are performed: if $i=n$, then the result $\pi=\pi^n$ is returned (work is finished); the set $F^i$ is generated through function **Get_F** (*see*, Figure 4 in ***Appendix A***); a recursive call of **PartOrderedPerm** with the parameters $\pi^{i+1}, \pi_{i+1}=f_j^i\in F^i, j\in J_{k^i}, k^i=Card\,F^i$ is made.

**procedure PartOrderedPerm** ( $\pi^i$ );
**if** $i = n$ **then print**( $\pi^i$ ); return;
$F^i$ :=**Get_F**( $\pi^i$ );
**for** $j = 1,2,...,card(F^i)$ **do PartOrderedPerm**( $\pi^{i+1} = (\pi_1, \pi_2, ..., \pi_i, f_j^i)$ );

**Fig. 1.** The main algorithm **PartOrderedPerm**

**Example 2.** Figure 2 illustrates an example of generating permutations from the elements $A = \{1,2,3,4,5\}$, for the assigned sets $D'(\pi) = \{1\}, \overline{D}'(\pi) = \{3,4\}$; the order of elements for position 2 hasn't been assigned. So, permutation $\pi \in P_5$ satisfies the given partially fixed order of elements $r(\pi)$ assigned for $k = 3$ positions $i \in \{1,3,4\}$ ) if and only if it satisfies one of the $2^{5-3-1} = 2$ complete orders of elements $\rho(\pi)$ that is identical to $r(\pi)$ for positions $i \in \{1,3,4\}$ .These two complete orders of elements descend in position 1 and ascend in positions 3, 4 and correspond to diagrams  and  .

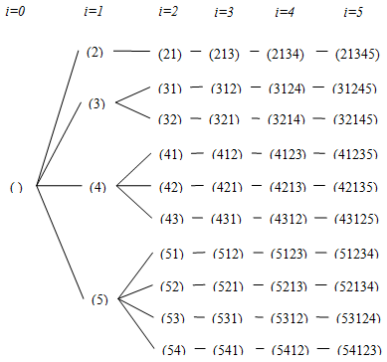



**Fig. 2.** The recursive tree demonstrating how the algorithm runs.

It should be noted that to generate other combinatorial sets with this algorithm, it is sufficient to identify the laws of constructing set $F^i$ for each level $i \in J_0^{n-1}$. Those laws are referred to in [8] for arrangements and combinations (either with repeated elements or without them).

### 3.1. Particular Cases of Permutations with a Partially Fixed Order

If the partially fixed order $r(\pi)$ has not been assigned to any position, the order of elements in permutations is not important at all. In this case, the generation results in the set $P_n$ of all permutations of $n$ elements.

If the partially fixed order $r(\pi)$ determines all positions $i \in \{1,2,...,n-1\}$, it becomes a complete order of elements $\rho(\pi)$. Then the described algorithm will generate permutations with a given signature [6].

## 4. Conclusions

In this paper, a special case of permutations – permutations with a partially fixed order of elements has been examined. This class of permutations generalizes permutations with a given signature described earlier. The number of such permutations has been evaluated.

An algorithm to generate such permutations has been suggested and its complexity has been evaluated (*cf.*, ***Appendix B***). The described algorithm also allows generating permutations with a given signature as a special

case of the described class of permutations but is more complex one than the algorithm described by Roeiants van Baronaigien and Ruskey [6].

It should be noted that the described **PartOrderedPerm** algorithm is **universal** and can be applied for generating other combinatorial sets. To do this, it is enough to assign the laws of constructing the set $F^i$ for a chosen combinatorial set.

## Acknowledgments

The authors would like to express their profound gratitude to the two anonymous reviewers whose valuable comments and suggestions have helped them considerably improve the text and the structure of the paper.

This research was partially financed by the SEP-CONACYT grant CB-2013-01-221676, Mexico, and by Tecnológico de Monterrey (ITESM), Campus Monterrey (Strategic Group of Numerical Methods and Industrial Engineering).

## References

1. Knuth, D.: The Art of Computer Programming, Vol. 4, Fasc. 2: Generating All Tuples and Permutations, Addison-Wesley Publishing Company, Boston, MA (2005).
2. Kreher, D.L., Stinson, D.R.: Combinatorial Algorithms: Generation, Enumeration, and Search, CRC Press, Taylor & Francis Group, London (1999).
3. Grebennik, I.V., Pankratov, A.V., & Baronov, A.V.: Packing *n*-dimensional parallelepipeds with the feasibility of changing their orthogonal orientation in an *n*-dimensional parallelepiped, Cybernetics and Systems Analysis, 46, 793-802 (2010).
4. Flajolet, R., Sedgewick, R.: Analytic Combinatorics, University Press, Cambridge (2009).
5. Stanley, R.P.: A Survey of Alternating Permutations (2009); www.math.mit.edu/rstan/papers/altperm.pdf
6. Roeiants van Baronaigien, D., Ruskey, F.: Generating permutations with given ups and downs, Discrete Applied Mathematics, 36(1), 57-65 (1992).
7. Abramson, M.: A simple solution of Simon Newcomb's problem, Journal of Combinatorial Theory Ser. A,18, 223-225 (1975).
8. Grebennik, I.V., Lytvynenko, O.S.: Generating combinatorial sets with given properties, Cybernetics and Systems Analysis, 48(6), 890-898 (2012).
9. Ruskey, F.: Combinatorial Generation, Dept. of Computer Science, University of Victoria, Canada, 1j-CSC 425/520 (2003).
10. Stanley, R.P.: Enumerative Combinatorics, Vol. 1, Wadsworth Inc., Belmont, CA (1986).

## 5. Appendix A

**Condition 3.** If, starting from position $i+1$, we require $m$ descents in a row, then there must exist at least $m$ elements of $A$ not belonging to $\pi^i$ and having values smaller than $\pi_{i+1}$:

$$\{i+1, i+2, ..., i+m\} \subseteq D'(\pi) \Rightarrow \exists L = \{a_{l_1}, a_{l_2}, ..., a_{l_m}\} \subseteq A: \forall a_{l_j} < \pi_{i+1},\ a_{l_j} \neq \pi_k, \quad (10)$$

where $j \in J_m$, $k \in J_i$, $m \in J_{n-1}$. If starting from position $i+1$, we require $m$ successive ascents, then there must exist at least $m$ elements of $A$ not belonging to $\pi^i$ and having values greater than $\pi_{i+1}$:

$$\{i+1, i+2, ..., i+m\} \subseteq \overline{D}'(\pi) \Rightarrow \exists L = \{a_{l_1}, a_{l_2}, ..., a_{l_m}\} \subseteq A: \forall a_{l_j} > \pi_{i+1},\ a_{l_j} \neq \pi_k, \quad (11)$$

where $\forall j \in J_m$, $k \in J_i$, $m \in J_{n-1}$. If the partially fixed order of elements $r(\pi)$ does not include position $i+1$, the following element $\pi_{i+2}$ can be either greater or smaller than $\pi_{i+1}$; and again, there are no limitations for $\pi_{i+2}$:

$$i+1 \notin D'(\pi), i+1 \notin \overline{D}'(\pi) \Rightarrow \pi_{i+1} \rho_{i+1} \pi_{i+2}, \rho_{i+1} \in \{<, >\}. \quad (12)$$

For a partial permutation $\pi^i$, there may exist several generating elements that can be selected as $\pi_{i+1}$. Let the set of such elements be denoted as $F^i$, $i \in J_{n-1}^0$. At a level $i \in J_{n-2}$, the set $F^i$ contains all the generating elements satisfying Eq. (6) through Eq. (12). However, the laws of forming the subset $F^i$ are different at the levels $i=0$ and $i=n-1$. Indeed, at level 0, conditions Eq. (6) – Eq. (9) are not verified because the permutation $\pi^0$ has no entries. In this case, the subset $F^0$ is constructed in the following way: if a given partially fixed order $r(\pi)$ requires $m$ descents at the beginning, then each element of the subset $F^0$ is to be such that there exist $m$ generating elements smaller than this element. Therefore, the subset $F^0$ consists of the latter $n-m$ elements of the generating set:

$$\{1,2,...,m\} \subseteq D'(\pi) \Rightarrow F^0 = \{a_{m+1}, a_{m+2}, ..., a_n\}. \quad (13)$$

If on the contrary, a given partially fixed order $r(\pi)$ requires $m$ ascents at the beginning, then each element of the subset $F^0$ has to be such that there exist $m$ generating elements greater than this element. Therefore, the subset $F^0$ consists of the former $n-m$ elements of the generating set:

$$\{1,2,...,m\} \subseteq \overline{D}'(\pi) \Rightarrow F^0 = \{a_1, a_2, ..., a_{n-m}\}. \quad (14)$$

If the partially fixed order of elements $r(\pi)$ does not include the position $i=1$, then the subset $F^0$ includes all the generating elements:

$$1 \notin D'(\pi), 1 \notin \overline{D}'(\pi) \Rightarrow F^0 = \{a_1, a_2, ..., a_n\}. \quad (15)$$

At the level $n-1$, the subset $F^{n-1}$ contains a single generating element not involved in $\pi^{n-1}$. Such an element should automatically satisfy the partially fixed order of elements $r(\pi)$ since its existence has been proved by conditions from Eq. (10) and Eq. (11) at the level $n-2$. Therefore, no condition is checked at the level $n-1$. It should be noted that at each level $i \in J_{n-2}^0$, the parameter $m$ has a value depending upon $r(\pi)$. For convenience, at each level $i \in J_{n-2}^0$, let the number of successive ascents/descents required in a partially fixed order $r(\pi)$ be denoted with $m^i$ starting from the next position $i+1$. Note that $m^i$ is determined only if the position $i$ is included into $D'(\pi)$ or $\overline{D}'(\pi)$, i.e., $m^i$ is not defined when the position $i+1$ is not assigned. The values of $m^i$ are determined in the following way. If a partially fixed order of elements $r(\pi)$ requires $m^i$ successive descents starting from the position $i+1$, then

$$i+1 \in D(\pi) \Rightarrow m^i = \max(t : i+t \in D(\pi)), t \in J_{n-i-2}. \quad (16)$$

Conversely, if a partially fixed order of elements $r(\pi)$ implies $m^i$ successive ascents starting from the position $i+1$, then

$$i+1 \in \overline{D}(\pi) \Rightarrow m^i = \max(t : i+t \in \overline{D}(\pi)), t \in J_{n-i-2}. \quad (17)$$

Now all values $m^i$ for a partially fixed order $r(\pi)$ can be calculated by the procedure that we call **Get_M.** Its input data are the cardinality $n$ of the set of generating elements and the partially fixed order of elements $r(\pi)$ associated with the sets $D'(\pi)$ and $\overline{D}'(\pi)$.

**procedure Get_M**

**for i:=0,1,...,n-2 do**

**if** $\{i+1\} \subseteq D'(\pi)$ **then** $m^i := \max(t : i+t \in D'(\pi))$

**else** $m^i := \max(t : i+t \in \overline{D}'(\pi))$

**Fig. 3.** Procedure **Get_M** calculating $m^i$.

Now let us describe the operation of procedure **Get_F** (Figure 4) that generates the set $F^i$ based on a partial permutation $\pi^i$. First, if $i = 0$, then by means of Eq. (13)–Eq. (15) (subject to $r(\pi)$), the set $F^0$ is created. On the other hand, if $i \neq 0$ then the set $F^i$ is generated by searching all the generating elements $a_j \in A$ not captured in $\pi^i$. The variable "condition2" has the value TRUE if the conditions Eq. (7)–Eq. (9) are satisfied and FALSE otherwise. Similarly, the variable "condition3" codes the validity of the conditions Eq. (10)–Eq. (12). It has been mentioned above that for the level $n-1$ no conditions are verified at all. All the elements $a_j \in A$ satisfying the above-mentioned constraints are added to the set $F^i$. Since the elements $a_j \in A$ are searched sequentially and the set $A$ is arranged in a strictly increasing order, the elements of $F^i$ are also arranged in a strictly increasing order. Input data for the procedure **Get_F** are: the cardinality $n$, sets $D'(\pi)$ and $\overline{D}'(\pi)$, the set $A$ of generating elements, a current partial permutation $\pi^i$, and the values $m^i, i \in J_{n-2}^0$ obtained through procedure **Get_M**.

**function Get_F**( $\pi^i$ );

$F^i := \varnothing$; $m := m^i$;

**if** $i = 0$ **then**:

**if** $1 \in D'(\pi)$ **then** $F^i := \{a_{m^0+1}, a_{m^0+2}, ..., a_n\}$;

**if** $1 \in \overline{D}'(\pi)$ **then** $F^i := \{a_1, a_2, ..., a_{n-m^0}\}$;

**if** $1 \notin D'(\pi), 1 \notin \overline{D}'(\pi)$ **then** $F^i := \{a_1, a_2, ..., a_n\}$;

**return** $F^i$;

**for** ( $a \in A, a \neq \pi_j^i$, $j = 1, 2, ..., i$ ):

**if (** $i \in D(\pi)$ **and** $a < \pi_i^i$ **) or (** $i \in \overline{D}'(\pi)$ **and** $a > \pi_i^i$ **) or** $i \notin D'(\pi), i \notin \overline{D}'(\pi)$ **then** condition2:=true

**else** condition2:=false

**if** $i+1 \in D'(\pi)$

**then** condition3:= $\exists L = \{a_{l_1}, a_{l_2}, ..., a_{l_{m^i}}\} \subseteq A : \forall a_{l_j} < a$, $a_{l_j} \neq \pi_k$, $\forall j \in J_{m^i}$, $\forall k \in J_i$;

**if** $i+1 \in \overline{D}'(\pi)$

**then** condition3:= $\exists L = \{a_{l_1}, a_{l_2}, ..., a_{l_{m^i}}\} \subseteq A : \forall a_{l_j} > a$, $a_{l_j} \neq \pi_k$, $\forall j \in J_{m^i}$, $\forall k \in J_i$;

**if** $i+1 \notin D'(\pi), i+1 \notin \overline{D}'(\pi)$ **then** condition3:=true;

**if** ( $i = n-1$ ) **or** (condition2 **and** condition3) **then** $F^i := F^i \cup a$;

**return** $F^i$;

**Fig. 4.** The function Get_F( $\pi^i$ ) forming the set $F^i$

# 6. Appendix B

**Evaluating the Complexity of the PartOrderedPerm Algorithm**

Basing upon the usual methods of evaluating a recursion algorithm's complexity (*cf.*, F. Ruskey [9]), we define the *complexity* of **PartOrderedPerm** algorithm as the quantity of changes in the interim data starting at the moment the algorithm is called up and finishing with its returning the final result. When **PartOrderedPerm** is called at each level $i \in J_{n-1}^0$, from the current permutation $\pi^i$ with the aid of the elements of the set $F^i$, a new entry $\pi_{i+1}$ is generated; after that **PartOrderedPerm** is called with the parameter $\pi^{i+1}$. Therefore, the algorithm complexity can be evaluated as a number of calls of **PartOrderedPerm** at all the levels $i \in J_{n-1}^0$.

**Theorem 1**. *The **PartOrderedPerm** algorithm constructs a backtracking tree (cf.,* [9]*) where each permutation π with a partially fixed order* $r(\pi)$ *serves as a leaf.*

*Proof.* If a tree leaf is a permutation $\pi$ of the length $n$, it means that all elements of this permutation $\pi^n$ satisfy the conditions Eq.(6)–Eq.(12) and for that reason $\pi^n$ is the permutation that satisfies $r(\pi)$. Let us demonstrate that this recursion tree has no deadlocks, i.e., such leaves that the length of $\pi^i$ is *less* than $n$.

A deadlock at a level $i \in J_{n-1}^0$ occurs when the set $F^i$ is empty, i.e., there are no generating elements that simultaneously satisfy the conditions Eq.(6)–Eq.(12). Therefore, by proving that $F^i$ contains at least one element at each level $i \in J_{n-1}^0$ we establish that the recursive tree of **PartOrderedPerm** does *not* have any deadlocks.

At the level $i$, there are always $n-i$ generating elements *not* involved in $\pi^i$. Thus, we can always find $n-i$ generating elements that satisfy the condition Eq.(6). Since the generating elements are distinct, each of $n-i$ generating elements not included in $\pi^i$ can be greater or smaller than $\pi_i$. In order to satisfy the conditions Eq.(7) and Eq.(8), a generating element has to be smaller or greater than $\pi_i$ depending on the partially fixed order $r(\pi)$.

From Eq.(10)–Eq.(11) it follows that, at the level $i-1$, the element $\pi_i$ is selected in such a manner that there exist at least $m^{i-1} \geq 1$ generating elements not involved in $\pi^{i-1}$, which are either greater or smaller than $\pi_i$ (depending on $r(\pi)$). The latter evidently implies that at the level $i$ there always exist no less than $m^{i-1} \geq 1$ generating elements that satisfy Eq.(7)–Eq. (8).

Let us denote the strictly increasing set of generating elements that satisfy the conditions Eq.(6)–Eq.(8) at the level $i$ by $C^i = \{a_{c_1}, a_{c_2}, ..., a_{c_p}\}$, $c_1 < c_2 < ... < c_p$, $\{c_1, c_2, ..., c_p\} \subseteq J_n$. For the elements of the set $C^i$ to be included in the set $F^i$, they have to satisfy the conditions Eq.(10)–Eq.(11).

An element $a_{c_k}, k \in J_p$ satisfies the conditions Eq.(10)–Eq.(11) if, and only if there exists $m^i$ generating elements from the set $C^i$ that are smaller or greater than $a_{c_k}$ (depending on $r(\pi)$). Making use of a somewhat modified terminology used by R. Stanley [10] we notice that, according to the given order of elements $r(\pi)$, the element $\pi_{i+1}$ can be:

a descent, if $\pi_i > \pi_{i+1} > \pi_{i+2}$, i.e. $\{i, i+1\} \subseteq D'(\pi)$;
a peak, if $\pi_i < \pi_{i+1} > \pi_{i+2}$, i.e. $i \in \overline{D}'(\pi), i+1 \in D'(\pi)$;
an ascent, if $\pi_i < \pi_{i+1} < \pi_{i+2}$, i.e. $\{i, i+1\} \subseteq \overline{D}'(\pi)$;
a valley, if $\pi_i > \pi_{i+1} < \pi_{i+2}$, i.e. $i \in D'(\pi)$, $i \in \overline{D}'(\pi)$.

In our case, for descents and peaks, starting from the position $i+1$, there may occur $m^i$ positions belonging to set $D'(\pi)$, i.e. $\{i, i+1, ..., i+m^i\} \subseteq D'(\pi)$ for a descent, and $i \in \overline{D}'(\pi), \{i+1, ..., i+m^i\} \subseteq D'(\pi)$ for a peak. Similarly, starting from the position $i+1$, for ascents and valleys, there may occur $m^i$ positions belonging to set $\overline{D}'(\pi)$, i.e. $\{i, i+1, ..., i+m^i\} \subseteq \overline{D}'(\pi)$ for an ascent and $i \in D'(\pi), \{i+1, ..., i+m^i\} \subseteq \overline{D}'(\pi)$ for a valley.

Thus, apart from Eq.(6), the conditions Eq.(7) and Eq.(10) have to be verified for the descents, Eq.(8) and Eq.(10) – for the peaks, Eq.(8) и Eq.(11) – for the ascents, Eq.(7) and Eq.(11) – for the valleys. For each of these four cases, the two remaining conditions are not verified.

It should be noted that in [10], descents and ascents are understood as the corresponding *elements* of a permutation, while in this paper, we mean *positions* belonging to permutation sets of descents or ascents, respectively.

If $\pi_{i+1}$ fixes ascents or descents, then the existence of $m^j = m^{i-1} - 1$ elements of the set $C^i$ that are greater or, on the contrary, smaller than $\pi_{i+1}$, is verified at the previous level $i-1$ in the conditions Eq.(10) or Eq.(9), respectively.

When $\pi_{i+1}$ occupies a peak, then the set $C^i$ contains unused generating elements that satisfy Eq.(8), i.e., are greater than $\pi_i$. To satisfy the relationship Eq.(10) for the element $a_{c_k}$, the existence of $m^i$ elements of the set $C^i$ that are smaller than $a_{c_k}$ is required. If the set $C^i$ is a singleton containing only one element $a_{c_1}$, then the remaining $n-i-1$ elements not involved in $\pi^i$ violate the restriction Eq.(10). In other words, all those elements are smaller than $\pi_i$ and hence smaller than $a_{c_1}$.

Next, if $C^i$ contains $p > 1$ elements, then there are $n-i-p$ elements not included in $\pi^i$ and not belonging to $C^i$. i.e., violating Eq.(8). If $n-i-p \geq m^i$ then Eq.(10) is true for all elements of $C^i$, since for every $a_{c_k}$ there always exist $n-i-p$ generating elements smaller than $\pi_i$ and not involved in $\pi^i$.

If $n-i-p < m^i$ then Eq.(10) does not hold for *all* elements of $C^i$ but is true only for those that have $m^i$ elements that are smaller than $a_{c_k}$ and not involved in $\pi^i$. Thus, the existence of $m^i$ elements that are smaller than $a_{c_k}$ is required, however, there are only $n-i-p$ of them beyond the set $C^i$. It means that there must be $m^i - (n-i-p)$ elements of $C^i$ that are smaller than $a_{c_k}$. Therefore, the condition Eq.(10) holds true for the latter $p - (m^i - (n-i-p)) = n-i-m^i$ elements of the set $C^i$. Moreover, $m^i$ is always smaller than $n-i$ because the number of relations among $n-i$ elements is clearly $n-i-1 \geq m^i$.

If $\pi_{i+1}$ is a valley, then the set $C^i$ contains unused generating elements that satisfy Eq.(7), i.e., are smaller than $\pi_i$. The element $a_{c_k}$ satisfies the condition Eq.(11) if there exist $m^i$ elements of the set $C^i$ that are greater than $a_{c_k}$.

If the set $C^i$ is a singleton containing only one element $a_{c_1}$, then the remaining $n-i-1$ elements not involved in $\pi^i$ violate the restriction Eq.(11). In other words, all those elements are greater than $\pi_i$ and hence greater than $a_{c_1}$. If $C^i$ contains $p>1$ elements, then there are $n-i-p$ of them not included in $\pi^i$ and not belonging to $C^i$, i.e., violating the condition Eq.(7). If $n-i-p \geq m^i$ then Eq.(11) is true for all elements of $C^i$, since for every $a_{c_k}$ there always exist $n-i-p$ generating elements greater than $a_{c_k}$ and not involved in $\pi^i$.

If $n-i-p < m^i$, then Eq.(10) does not hold for *all* elements of $C^i$ but is true only for those that have $m^i$ elements that are smaller than $a_{c_k}$ and not involved in $\pi^i$. Thus, it should exist $m^i$ elements being greater than $a_{c_k}$, but there are only $n-i-p$ of them outside the set $C^i$. It means that there must be $m^i-(n-i-p)$ elements of $C^i$ that are greater than $a_{c_k}$. Therefore, the relationship Eq.(10) is valid for the former $p-(m^i-(n-i-p)) = n-i-m^i$ elements of the set $C^i$.

To satisfy Eq.(9), the generating element can be either greater or smaller than $\pi_i$. Therefore, all generating elements satisfy Eq.(9). The same reasoning allows stating that all generating elements satisfy (12).

We have just shown that the set $F^i$ has at least one element at each level $i \in J_{n-1}^0$, which means the absence of deadlocks in the recursion tree grown by **PartOrderedPerm** and thus proves the theorem.

Now following the terminology from [9] and having established the validity of Theorem 1, it can be concluded that **PartOrderedPerm** algorithm belongs to the BEST (Backtracking Ensuring Success at Terminals) class. In other words, it is an algorithm of the backtracking type, but every leaf of the backtracking tree is an object of the desired kind [9].

**Proposition 1**. ***PartOrderedPerm*** *algorithm is a ranking algorithm, as permutations* $\pi$ *are generated in the lexicographical order*.

*Proof.* As mentioned above, the elements of the set $F^i$ strictly increase: $f_1^i < f_2^i < ... < f_{k^i}^i$, $f_j^i \in F^i$, $j \in J_{k^i}$, $k^i = Card\, F^i$. The algorithm inserts the elements of $F^i$ (starting from the first one) into the current partial permutation $\pi^i$, and then calls **PartOrderedPerm** again. This means that the recursive calls of **PartOrderedPerm** are made first for the smallest elements of $F^i$ and subsequently are applied to the ever growing elements of $F^i$, which finally leads to the lexicographic ordering of generated permutations. The proof is complete.

Let us denote the total number of permutations of *n* elements with a given partially fixed order of elements by $N$.

**Proposition 2.** *The complexity of* ***PartOrderedPerm*** *algorithm is* $O(nN)$.

*Proof.* As mentioned above, we decided to evaluate the complexity of the algorithm by the number of calls of **PartOrderedPerm** through all the levels $i \in J_{n-1}^0$. Since at each level $i \in J_{n-1}^0$, **PartOrderedPerm** adds one element in $\pi^i$ and there can be no deadlocks (*see*, Theorem 1), the procedure **PartOrderedPerm** is called no more than $n$ times to generate each of $N$ permutations. Therefore, the total number of calls of **PartOrderedPerm** at all the levels $i \in J_{n-1}^0$ does not exceed $nN$, which finishes the proof.