COMMENTATIONES MATHEMATICAE UNIVERSITATIS CAROLINAE
23,2 (1982)

EXTENSIONS OF k-SUBSETS TO k+1-SUBSETS – EXISTENCE VERSUS CONSTRUCTABILITY

S. POLJAK, D. TURZÍK, P. PUDLÁK

Abstract: Our aim is to look for algorithms which construct objects whose existence is proved nonconstructively. We present two algorithms, one that for any given k-subset of a set X finds a disjoint k-subset of X so that distinct subsets have distinct images, and one that extends any given k-subset of a set X to a k+1-subset of X so that distinct k-subsets have distinct extensions. We discuss some relations between decision and construction problems.

Key words: Algorithm, polynomial time, finite set.

Classification: 68C05

Introduction. Our aim is to look for algorithms which construct objects whose existence is proved nonconstructively. In § 1 we consider the following two problems concerning subsets.

1. Let X be a set of cardinality n and $k \leq n/2$. For any given k-subset A find a disjoint k-subset B, denoted by B = = DIS(A,X), so that distinct subsets have distinct images, i.e. $A \neq A'$ implies $DIS(A,X) \neq DIS(A',X)$.

2. Let X be a set of cardinality n and $k < n/2$. Extend any given k-subset A to a k+1-subset B, denoted by B = EXT(A,X), so that distinct k-subsets have distinct extensions, i.e. $A \neq A'$ implies $EXT(A,X) \neq EXT(A',X)$.

Using the König-Hall theorem one can easily prove the existence of such mappings DIS and EXT. We present two algorithms, DIS and EXT which for a given subset A construct DIS(A,X) and EXT(A,X) in polynomial time.

In § 3 we specify a class of problems - we call them purely constructive - for which the decision problem is easy while the constructive one might be hard. Two problems about Hamiltonian cycles discussed in § 2 are examples of purely constructive problems.

§ 1. A construction of a mapping φ can be understood in two ways.

(a) A construction of the list of all pairs $(x, \varphi(x))$,

(b) a procedure which for a given x constructs $\varphi(x)$.

If we wanted to obtain a list of all pairs $(A, \varphi(A))$ in the case when φ is either DIS(-,X) or EXT(-,X), we could do it well by using the matching algorithm ([5]). But such a list might be of size exponential in n, e.g. for $k \sim n/3$. The latter approach will be more convenient in the following case.

Given an input sequence $A_1, A_2, \ldots, A_r$ of (not necessarily all) subsets of X we are to construct the sequence $\varphi(A_1)$, $\varphi(A_2), \ldots, \varphi(A_r)$ provided when dealing with A_i we do not know the other members A_j, $j > i$.

We present two algorithms DIS and EXT which for a given k-subset A construct DIS(A,X) and EXT(A,X), respectively, in $O(k^2)$ steps. x)

----

x) We count writing a number, comparing two numbers etc. as a single step as in the model of RAM's.

<u>Description of the algorithm DIS</u>. Let X be a set of cardinality n and A its subset of cardinality $k \leq n/2$. Suppose $X = \{0,1,\dots,n-1\}$ and $A = \{a_1,\dots,a_k\}$. Suppose the set X forms a cycle. The image DIS(A,X) of A is constructed as "a shadow" of A.

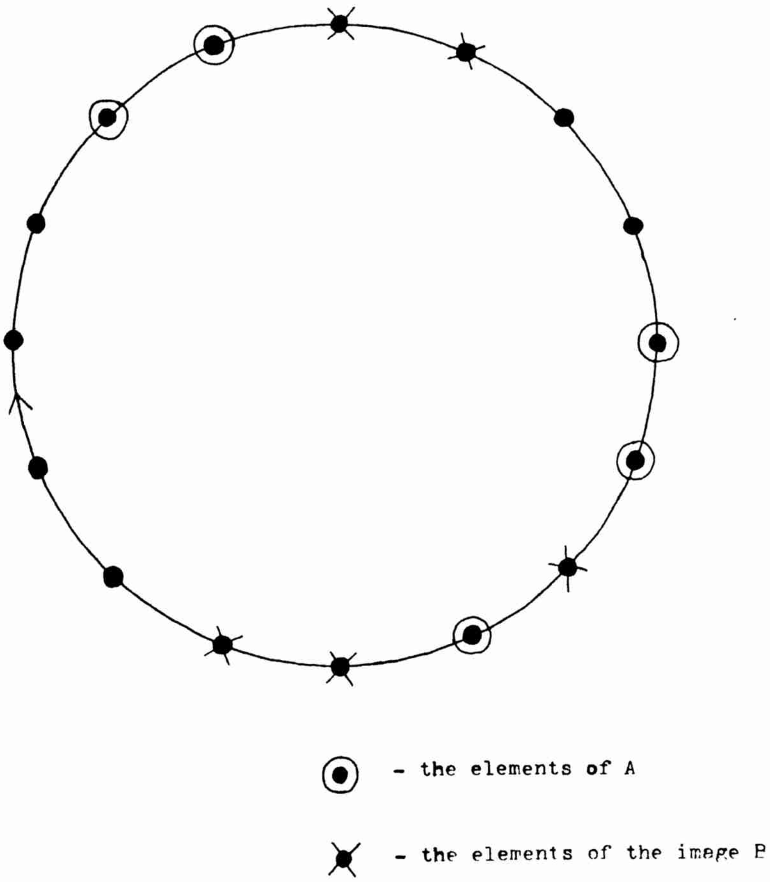


Fig. 1

Formally, the procedure DIS constructs an output sequence $(b_1, b_2, \dots, b_k)$ for an input sequence $(a_1, a_2, \dots, a_k)$ as follows.

Procedure DIS(A,X);

$B := \emptyset$ (the empty set);

For $j := 1$ to k do

Begin

$b_j := a_j$;

Do $b_j := b_j + 1 \bmod n$ until $b_j \notin A \cup B$;

$B := B \cup \{b_j\}$;

end;

$DIS((a_1, \dots, a_k), X) := (b_1, \dots, b_k)$;

<u>Lemma</u>. Let $(a_1', \dots, a_k')$ be a permutation of $(a_1, \dots, a_k)$ and $DIS((a_1', \dots, a_k'), X) = (b_1', \dots, b_k')$. Then $\{b_1', \dots, b_k'\} = \{b_1, \dots, b_k\}$.

Proof. As every permutation can be decomposed into translations of pairs of consecutive members, it is sufficient to prove the lemma only for the case when $(a_1', \dots, a_k') = (a_1, \dots, a_{i-1}, a_{i+1}, a_i, a_{i+2}, \dots, a_k)$ for $i = 1, 2, \dots, k-1$.

Clearly, $b_j' = b_j$ for $j = 1, \dots, i-1$. Then either $b_i' = b_i$ and $b_{i+1}' = b_{i+1}$, or $b_i' = b_{i+1}$ and $b_{i+1}' = b_i$. Hence $\{b_1, \dots, b_{i+1}\} = \{b_1', \dots, b_{i+1}'\}$, and then again $b_j' = b_j$ for $j = i+2, \dots, k$. □

Thus, the output set B does not depend on the ordering of the input set A. In the following DIS(A,X) will mean the output set (without ordering).

<u>Theorem 1</u>. The algorithm DIS satisfies

(1) $|DIS(A,X)| = k$

(2) $A \cap DIS(A,X) = \emptyset$

(3) $A \neq A' \Rightarrow DIS(A,X) \neq DIS(A',X)$

for all k-subsets A, A' of a set X, $k \leq |X|/2$. Moreover, DIS(A,X) is constructed in $O(k^2)$ steps.

Proof. The (1),(2) and the number of steps are easy to check. We prove (3). Consider a procedure DIS^{-1} defined in the same way as DIS but $b_j := b_j - 1 \bmod n$. Clearly DIS and DIS^{-1} are inverse. □

Description of the algorithm EXT. Let X be a set of cardinality n and A its subset of cardinality $k < n/2$. Suppose $X = \{1,2,\dots,n\}$ and $A = \{a_1,a_2,\dots,a_k\}$ with $a_1 < a_2 < \dots < a_k$.

1. Find an integer t(A) such that

$t(A) = \max \{t \mid |A \cap \{1,2,\dots,2t+1\}| = t\}$.

(It may happen $t(A) = 0$.)

2. Set $Y = \{1,2,\dots,2t(A)+1\}$ and apply the algorithm DIS to the input $(A \cap Y, Y)$. Then the set $DIS(A \cap Y,Y)$ is a set of cardinality t(A) disjoint to $A \cap Y$. Thus $Y \setminus DIS(A \cap Y,Y)$ is a set of cardinality $t(A)+1$ containing $A \cap Y$.

3. Set $EXT(A,X) = A \cup (Y \setminus DIS(A \cap Y,Y))$ which is a set of cardinality k+1 containing A.

```
    Procedure EXT(A,X);
    t:= k;
(4) While a_t > 2t+1 do t:= t-1;
    Y:= {1,2,...,2t+1};
    EXT(A,X):= A ∪ (Y \ DIS(A ∩ Y,Y));
```

Theorem 2. The algorithm EXT satisfies

(5) $|\mathrm{EXT}(A,X)| = k+1$

(6) $A \subset \mathrm{EXT}(A,X)$

(7) $A \neq A' \Longrightarrow \mathrm{EXT}(A,X) \neq \mathrm{EXT}(A',X)$

for all k-subsets A, A′ of a set X, $k < |X|/2$. Moreover, EXT(A,X) is constructed in $O(k^2)$ steps.

Proof. The (5),(6) and the number of steps are easy to check. Clearly t(A) is the integer t constructed by (4). We prove (7).

Let A and A′ be two distinct k-subsets of X. Assume $t(A') \leq t(A)$. We distinguish two cases.

(i) $t(A') = t(A)$. Put $Y = \{1,2,\dots,2t(A)+1\}$. Then either $A \setminus Y \neq A' \setminus Y$ or $A \cap Y \neq A' \cap Y$. In the former case $\mathrm{EXT}(A,X) \neq \mathrm{EXT}(A',X)$ as the added elements belong to Y. In the latter case $\mathrm{DIS}(A \cap Y, Y) \neq \mathrm{DIS}(A' \cap Y, Y)$ by (3), and hence (7).

(ii) $t(A') < t(A)$. Set Y as above.

Then

(8) $|\mathrm{EXT}(A,X) \cap Y| = t(A)+1,$

and

(9) $|\mathrm{EXT}(A',X) \cap Y| \leq t(A),$

as $|A' \cap Y| < t(A)$. Thus, by (8) and (9), the sets EXT(A,X) and EXT(A′,X) have distinct intersection with Y, and (7) follows. □

<u>Remark 1</u>. The k+1-subsets B, the extensions of k-subsets constructed by the algorithm EXT, can be recognized as those satisfying

$|B \cap \{1,2,\dots,2t+1\}| = t+1$ for some $t = 0,1,\dots,[\frac{n-1}{2}]$.

<u>Remark 2</u>. Put $P_k(X) = \{A \subset X \mid |A| = k\}$. Define bipartite graphs G_1 and G_2 as follows.

$V(G_1) = P_k \times \{0,1\}$,

$\{(A,0),(B,1)\}$ is an edge of G_1 if $A \cap B = \emptyset$,

$V(G_2) = P_k(X) \cup P_{k+1}(X)$,

$\{A,B\}$ is an edge of G_2 if $A \subsetneq B$.

Hence G_1 is a regular bipartite graph of degree $\binom{n-k}{k}$, and G_2 is a bipartite graph with vertices of degree n-k in P_k and k+1 in P_{k+1}. It follows from the König-Hall theorem that G_1 has a perfect matching iff $k \leq n/2$, and P_k can be matched into P_{k+1} in G_2 iff $n-k \geq k+1$ ($\Longleftrightarrow k < n/2$). (See [2], Chapter 7, Corollary 2 of Theorem 2.)

Remark 3. Let G_1 and G_2 be as above. Using parallel processing, the algorithms DIS and EXT construct a maximal matching in G_1 and G_2, in $O(\log^2|G_1|)$ and $O(\log^2|G_2|)$ steps, respectively.

Remark 4. Let n,k,r be positive integers satisfying $2k + r \leq n$. Define $EXT^{(r)}$ in the same way as EXT but

(4′) While $a_k > 2t+r$ do t:= t-1;

instead of (4). Then $EXT^{(r)}$ constructs a one-to-one extension of k-subsets to k+r-subsets.

We were informed that related questions were studied in [8], which yields another algorithm for the mapping EXT.

§ 2. The approach of the previous section can be characterized as follows. We were able to prove existence easily and we tried to find an algorithm. The same situation has appeared also in other problems. For example, Chvátal [4] proved that any graph G with n vertices and degrees $d_1 \leq d_2 \leq \ldots \leq d_n$ satisfying

$$(\ast) \quad d_k \leq k < n/2 \Rightarrow d_{n-k} \geq n-k$$

must have a Hamiltonian cycle. An algorithm finding a Hamiltonian cycle for graphs satisfying $(\ast)$ in polynomial time was given later in [3].

In 1946 C.A.B. Smith proved the following theorem.

In a simple regular graph of degree 3, the number of Hamiltonian cycles that contain a given edge is even.

A nonconstructive proof of this theorem, based on counting modulo 2 the number of cycles, is in ([2], Chapter 10, Theorem 2). This theorem suggests the following problem. Given a triple (G,e,C), where G is a 3-regular graph, and C a Hamiltonian cycle of G containing the edge e, construct another Hamiltonian cycle containing e. We do not know whether there is a polynomial algorithm for this problem. Thomason [7] suggested the following algorithm, but it is not clear how many steps the algorithm requires in the worst case.

1. Let $C = (x_1,\dots,x_n)$ be the given Hamiltonian cycle containing the given edge $e = (x_1,x_2)$. A sequence $P_o, P_1 \dots$ is constructed until P_k forms a Hamiltonian cycle for some $k > 0$.

$P_o = (x_1,x_2,\dots,x_n)$

$P_1 = (x_1,\dots,x_i,x_n,x_{n-1},\dots,x_{i+1})$ where $(x_n,x_i) \quad E(G)$, $x_i \neq x_1, x_{n-1}$.

...

If $P_j = (y_1,\dots,y_n)$,

put $P_{j+1} = (y_1,\dots,y_i,y_n,y_{n-1},\dots,y_{i+1})$ where (y_n,y_i) is the only edge incident to y_n which belongs neither to

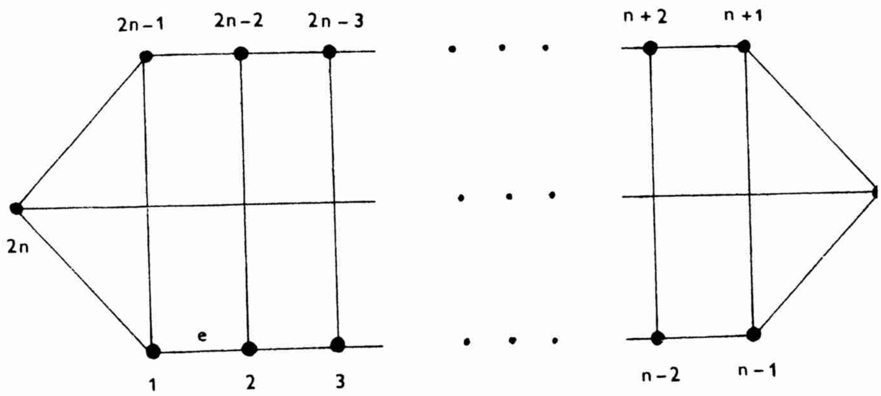
P_j nor to P_{j-1}.

We have found a family of graphs $\{G_n\}$ for which the algorithm constructs just $(n-1)^2+2$ paths. The graphs G_n are defined by

$$V(G_n) = \{x_1, x_2, \ldots, x_{2n}\}$$

$$E(G_n) = \{(x_i, x_{i+1}) \mid i = 1, \ldots, 2n-1\} \cup \{(x_{2n}, x_1)\} \cup$$
$$\cup \{(x_i, x_{2n-i}) \mid i = 1, \ldots, n-1\} \cup \{(x_n, x_{2n})\},$$

$C = (x_1, x_2, \ldots, x_{2n})$ and $e = (x_1, x_2)$.



G_n

Fig. 2

§ 3. In the theory of NP-problems usually only the existence of decision algorithms is investigated. In practice, however, it is more important to have an algorithm for a related construction problem. Namely, each NP-problem can be represented in the following form:

Given x determine whether there is y such that $R(x,y)$; where $R(x,y)$ is some polynomial time decidable relation and such that the size of y is bounded by a polynomial of the size of x.

Then the construction problem is: Given x, construct y such that $R(x,y)$ if there exists at least one such y.

It is known [6] that for NP-complete problems the existence of a polynomial time decision algorithm is equivalent with the existence of a polynomial time construction algorithm. This is a reason why most investigations deal only with the simpler concept of decision problems.

Consider for example Hamiltonian graphs. Then the relation $R(x,y)$ means x is a graph and y is a Hamiltonian cycle in x. Clearly, $R(x,y)$ can be decided in polynomial time. Suppose a decision algorithm for Hamiltonian graphs is given. Then in order to construct a Ham. cycle in a Ham. graph we can use the following simple procedure. Take an edge e in G and use the algorithm to test whether G–e is Hamiltonian. If not, try another edge. If you find an edge f such that G–f is Ham., repeat the procedure with G replaced by G–f until the remaining edges form a Ham. cycle. For general NP-complete problems the proof is very similar.

The situation is different for the two examples discussed in the preceding section. The structure of these problems is the following: We are given a polynomial time decidable predicate $S(x)$ such that for each x

$$S(x) \Rightarrow \exists y \; R(x,y),$$

and we need an algorithm which for x such that $S(x)$ constructs y such that R(x,y). We suggest to call problems of this structure *purely constructive* since their decision problems are by definition easy.

In the first example R(x,y) means y is a Ham. cycle in a graph x and S(x) is the Chvátal´s condition. This condition can be tested in polynomial time and his theorem [4] assures that each graph satisfying the condition is Hamiltonian. In the second example R(x,y) means x = (G,e,C), where G is a graph, e is an edge and C is a Ham. cycle in G containing e, and y is a Ham. cycle in G distinct from C and containing e.

On the other hand, one cannot show that maximal clique problem is purely constructive since no polynomial time algorithm is known for decision whether given clique is of maximal cardinality in a given graph.

There are two extreme possibilities for purely constructive problems:

I. For each purely constructive problem there is a polynomial time algorithm.

II. There is a purely constructive problem such that each construction algorithm for it is NP-hard.

We believe that the truth is somewhere inbetween (i.e. neither I. nor II. is true). Let us note that I. implies NP $\cap$ coNP= = P, and NP = coNP implies II. It was shown in [1] that one cannot prove or disprove P=NP, NP=coNP, NP coNP=P, and some other statements using methods that allow relativization. The same is true about the statements I. and II., thus they are probably very difficult, too.

Appendix. It is easy to modify the algorithm DIS so that it runs in O(k) steps. This gives us an algorithm for EXT which runs in O(k) steps as well.

Let $X = \{0,1,\dots,n-1\}$ and $A = \{a_1,a_2,\dots,a_k\} \subset X$ such that $a_1 < a_2 < \dots < a_k$, $k \leq n/2$.

```
     Procedure DIS(A,X);
     Begin
     B:= Ø (the empty set);
     j:= 0;
L1:  j:= j+1;
     p:= 1;
     i:= a_j;
L2:  i:= i+1 mod n;
     if i ∈ A then
        begin
        p:= p+1;
        j:= j+1;
        go to L2;
        end;
     if i ∈ B then go to L2;
     if p > 0 then
        begin
        p:= p-1;
        B:= B ∪ {i};
        go to L2;
        end;
     if j < k then go to L1;
     EXT(A,X):= B;
     end of procedure;
```

References

[1] T. BAKER, J. GILL and R. SOLOVAY: Relativizations of the P=?NP question, SIAM J. Comp. Vol. 4(1975), 431-442.

[2] C. BERGE: Graphs and Hypergraphs, North-Holland, Amsterdam (1973).

[3] J.A. BONDY and V. CHVÁTAL: A method in graph theory, Discrete Mathematics 16(1976), 111-135.

[4] V. CHVÁTAL: On Hamiltonian's ideals, Journal of Combinatorial Theory 12(1972), 163-168.

[5] J.E. HOPCROFT and R.M. KARP: An $n^{5/2}$ Algorithm for Maximal Matchings in Bipartite Graphs, SIAM J. Comp. Vol. 2(1973), 225-231.

[6] C.P. SCHNORR: Optimal algorithms for self-reducible problems, Automata, Languages and Programming 1976, Eds.: S. Michaelson and R. Milner, University Press Edinburgh, 322-337.

[7] A.G. THOMASON : Hamiltonian cycles and uniquely edge colourable graphs, Advances in Graph Theory B. Bollobás, ed., Annals of Discrete Mathematics 3(1978), 259-268.

[8] C. GREENE, D.J. KLEITMAN: Proof techniques in the theory of finite sets. Studies in Combinatorics, pp. 22-79, MAA Studies in Math. 17, Math. Assoc. America, Washington, D.C., 1978.

S. Poljak: Technical University, Faculty of Civil Engineering, Department of Economics, Thákurova 7, Praha 6, Czechoslovakia

D. Turzík: The University of Chemical Technology, Department of Mathematics, Suchbátarova 5, Praha 6, Czechoslovakia

P. Pudlák: Czechoslovak Academy of Sciences, Mathematical Institue, Žitná 25, Praha 1, Czechoslovakia

(Oblatum 21.9. 1981)