PETER RUŽIČKA
JURAJ WACZULÍK

Pebbling dynamic graphs in minimal space

Informatique théorique et applications, tome 28, nº 6 (1994), p. 557-565.

<http://www.numdam.org/item?id=ITA_1994__28_6_557_0>

Informatique théorique et Applications/Theoretical Informatics and Applications
(vol. 28, n° 6, 1994, p. 557 à 565)

PEBBLING DYNAMIC GRAPHS IN MINIMAL SPACE (*)

by Peter Ružička [1] and Juraj Waczulík [1]

Communicated by R. Cori

Abstract. – *Pebble game on dynamic graphs is studied as an abstract model of the incremental computations. Extreme explosion of the time complexity is related to small changes in the size of computational graphs. A class of dags $\{G_n\}$ of the size n pebbleable in space $O(n^{1/3})$ is exhibited such that the standard pebble game requires superpolynomial time in n under the restriction that the minimal space is used. Moreover, by deleting just one edge in $\{G_n\}$ a new class of graphs is obtained for which standard pebble game needs only polynomial time in n also in the case when the minimal space is used.*

Résumé. – *On étudie un jeu sur les graphes dynamiques qui constitue un modèle théorique de calcul incrémental. Une explosion considérable de la complexité temporelle a lieu pour de faibles modifications du graphe sur lequel on calcule. Ainsi on construit une classe de graphes acycliques $\{G_n\}$ de taille n que l'on peut traiter avec une complexité en espace en $O(n^{1/3})$ et qui nécessite un temps superpolynomial en n si l'espace minimal est utilisé. De plus, en supprimant simplement un arc dans $\{G_n\}$ une nouvelle classe est obtenue pour laquelle le jeu demande simplement un temps polynomial même si l'on utilise un espace minimal.*

INTRODUCTION

Pebbling directed acyclic graphs (shortly dags) introduce an important paradigm in programming. Dags present an abstract computational structure of a given problem and pebbling performs the computation of this problem on the computational graph. Pebbling mediates us to model various types of computations and enables us to investigate the relationship between space and time complexity of these computations by means of a simple combinatorial game played with pebbles on dags. There are well known exploitations of pebbling in gaining time-space trade offs for concrete computational problems in various areas of computer science, such as compilers, database systems, parallel and distributed systems, programming methodology and others. In the theory of pebbling well known results have been obtained

(*) Received September 1992, accepted June 1993.

[1] Institute of Computer Science, Comenius University, 84215 Bratislava, Slovakia.

concerning space complexity and time-space trade offs for pebbling special classes of directed acyclic graphs. A survey of principal classical results and basic applications of pebbling is given by Pippenger (1980). Recent pebbling results are presented by Wilber (1988) and Venkateswaran, Tompa (1989).

Recently much effort has been devoted to the investigation of the incremental computations. We propose to study pebble game on dynamic graphs, acting as a model of incremental computations. From the theoretical point of view we are interested in the relationship between space or time complexity of pebbling and small incremental/decremental changes of graphs.

In this paper we concentrate on space optimal pebbling. We characterize extreme changes in the time complexity with respect to small changes in the size of computational graphs. We present a class of dags $\{G_n\}$ of the size n which requires superpolynomial time in n for pebbling when the minimal space is used. But by deleting just one edge in $\{G_n\}$ a new class of graphs is obtained which requires only polynomial time for pebbling in the minimal space.

1. BASIC NOTIONS

Throughout this paper we consider only directed acyclic graphs (dags) with in-degree bounded by 2. An input (an outpout) in a dag is a vertex with in-degree (out-degree) equal to 0. If a dag has exactly one output, then this vertex is called the root of the dag and the dag is called rooted. The size of a dag G (denoted as *size* (G)) is equal to the number of its edges.

Standard pebble game is performed with pebbles on dags using following rules:

S-1: A pebble can be removed from an arbitrary vertex of the graph.

S-2: If all direct predecessors of a vertex v are covered by pebbles, then a pebble can be layed also on the vertex v.

S-3: If all direct predecessors of a vertex v are covered by pebbles, then a pebble can be moved from some predecessor of the vertex v onto the vertex v.

A configuration is a subset of vertices, comprising just those vertices that have pebbles on them. A configuration is empty if there is no pebble on any vertex. A transition is an ordered pair of configurations (C_1, C_2), where C_2 follows from C_1 according to one of the rules S-1, S-2, S-3. A computation is a sequence of configurations $C_1, C_2, \ldots, C_m$, where (C_k, C_{k+1}) for $1 \leqq k < m$ is a transition. A complete computation is one that begins and ends with the empty configuration and in which every vertex appears in some configuration. The time T is the number of transitions for complete computations and the space S is the maximum number of vertices in any configuration for complete computations.

A dag G is called open in a configuration C if and only if there is a path in G from an input to an output such that it does not contain a vertex in C. Otherwise G is called closed in C.

2. TIME RESULTS FOR THE MINIMAL SPACE PEBBLING ON DYNAMIC GRAPHS

The influence of small changes (performed e.g. by inserting or deleting an edge) in computational graphs onto the time complexity of standard pebbling is studied. As it has been previously noticed, we are interested only in the space optimal pebbling.

First of all we determine the space complexity of standard pebbling in the case when there is only a little change in the underlaying dag.

PROPOSITION 2.1: *Inserting an edge to an arbitrary dag can increase the space complexity of standard pebbling by at most* 1.

The proposition directly follows from the fact that if a new dag $\hat{G}$ is obtained from the original dag G by adding an edge (v, u) to G, then a complete computation on $\hat{G}$ can be constructed from a complete computation on G by placing a permanent pebble on the vertex v.

Now, we present an upper bound on time complexity for pebbling a graph of size n, which can be pebbled with $S(n)$ pebbles.

PROPOSITION 2.2: *Let G_n be a rooted dag with n vertices and let $S(n)$ be the number of pebbles sufficient for pebbling G_n. Then for the time $T(n)$ needed to pebble the graph G_n with $S(n)$ pebbles it holds*

$$T(n) = 2^{O(S(n) \cdot \log n / S(n))}.$$

Proof: Each complete computation on rooted dags can be transformed onto the complete computation without repeated configurations.

Consider a complete computation with $S(n)$ pebbles in the form starting with $S(n)$ applications of the rule S-2, following with a sequence of applications either of the rule S-3 or of the rule S-1 followed by the rule

S-2 and ending with $S(n)$ applications of the rule S-1. The worst-cast time complexity of these three phases of computation is estimated as follows

$$T(n) \leqq S(n) + 2 \,.\binom{n}{S(n)} + S(n).$$

Applying Stirling formula $n! \geqq \sqrt{2\pi n}\,(n/e)^n$ one can obtain the time estimation

$$T(n) \leqq \frac{\sqrt{2}}{\sqrt{\pi S(n)}} \cdot 2^{S(n) \log e \cdot n/S(n)} + 2 \,. S(n)$$

and hence the asymptotic time-space trade-off formula is in the form

$$T(n) = 2^{O(S(n) \cdot \log n/S(n))}. \quad \square$$

In a special case when a graph can be pebbled with a constant number of pebbles the following upper bound on time complexity can be derived.

CONSEQUENCE 2.3: *Let G_n be a rooted dag with n vertices and let k be a constant number of pebbles necessary and sufficient to pebble G_n. Then for the time $T(n)$ needed to pebble the graph G_n using k pebbles it holds*

$$T(n) = O(n^k).$$

Now, we present a lower bound on the time complexity of the minimal space pebbling up to space $O(n^{1/3})$ which is asymptotically tight to the previously given upper bound. Basic to all graphs discussed later in this paper is the pyramidal graph $P_k = (V, E)$, where

$$V = \{u_{i,j} \mid 1 \leqq i \leqq j \leqq k\}$$
$$E = \{(u_{i,j},\, u_{s,t}) \mid s = i + 1,\ t \in \{j,\, j + 1\},\ 1 \leqq i \leqq j,\ s \leqq t \leqq k\}$$

LEMMA 2.4 (Cook, 1974): *Each complete computation of the standard pebble game on the pyramidial graph P_k needs at least k pebbles.*

Consider 3 graph components P_i, R_j, H_l, where P_i is a pyramidal graph with i inputs, 1 output and $1/2\, i\,(i + 1)$ vertices, R_j is a chain graph with 1 input, 1 output and j vertices and H_l is a combination graph with 1 input, 2 outputs and $l\,(l + 2)$ vertices. A pyramid P_i is a fragment of rectangular grid. A chain R_j is a sequence of j vertices, where there is an edge oriented from the k-th vertex to the $(k + 1)$-st one. A combination graph H_l consists of two pyramids P_l and a chain $R_{2\,l}$, where the i-th vertex of $R_{2\,l}$ for $1 \leqq i \leqq l$

is connected with oriented edge to the i-th input vertex of the first pyramid and the j-th vertex of R_{2l} for $l+1 \leqq j \leqq 2l$ is connected with oriented edge to the $(j-l)$-th input vertex of the second pyramid.

A class of dags $\{G_{k,m} \mid k \geqq 1,\ m \geqq 2\}$ is defined as follows. $G_{k,m}$ is constructed from k levels $U_1^m, \ldots, U_k^m$, where the i-th level U_i^m for $1 \leqq i \leqq k$ consists of the serial connection of three components P_i, R_m, H_i (*i.e.*, the output of P_i is connected with oriented edge to the input of R_m and the output of R_m is connected to the input of H_i). The graph $G_{k,m}$ is completed using the following connections of levels U_i^m and U_{i+1}^m for $1 \leqq i < k$. The first (second) output of U_i^m is connected with oriented edges to all odd (even) vertices in the components R_m of the level U_{i+1}^m. See the Figure 1.

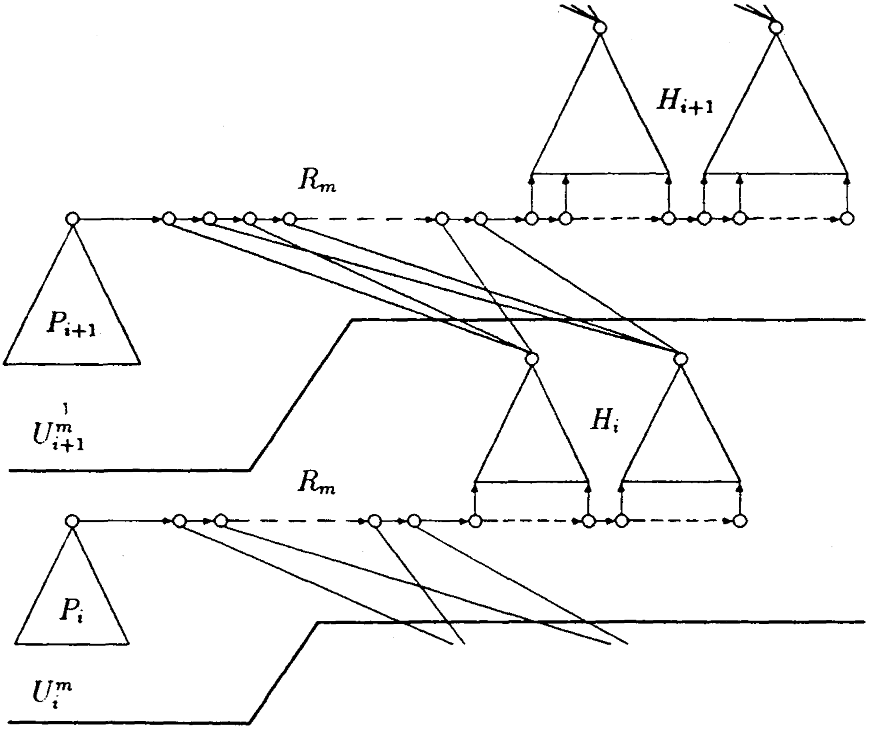

Figure 1. – The i-th and $(i+1)$-st level of the graph $G_{k,m}$.

PROPOSITION 2.5: *The class of graphs $\{G_{k,m} \mid k \geqq 1,\ m \geqq 2\}$ satisfies two conditions:*

(i) *the size of $G_{k,m}$ is equal to $\Theta(k^3 + k \,.\, m)$;*

(ii) *the minimal number of pebbles needed for complete computation on the graph $G_{k,m}$ is equal to k.*

Proof: Case (i). The size of the graph $G_{k,m}$ fulfills the following recurrent relation

$$\begin{aligned} size\,(G_{k,m}) &= a_1 \,.\, m + size\,(G_{k-1,m}) \\ size\,(G_{1,m}) &= a_2 \,.\, m \end{aligned}$$

for k in the range $\omega\,(1)$ to $O\,(\sqrt{m})$ and some constants a_1, $a_2 > 1$ and

$$\begin{aligned} size\,(G_{k,m}) &= a_3 \,.\, k^2 + size\,(G_{k-1,m}) \\ size\,(G_{1,m}) &= a_2 \,.\, m \end{aligned}$$

for k in the range $\Omega\,(\sqrt{m})$ to $m^{O\,(1)}$ and a constant $a_3 > 1$.

The solution is in the form

$$size\,(G_{k,m}) = \Theta\,(k^3 + k \,.\, m)$$

Case (ii): The proposition follows directly from the Lemma 2.4. □

Consider now the k-th level of the graph $G_{k,2m}$. Denote by u the output of the first component P_k of the k-th level U_k^{2m} and by $x_0, x_1, \ldots, x_{2m-1}$ the sequence of vertices in the second component R_{2m} of the k-th level U_k^{2m} and by α_0, α_1 the output vertices of the $(k-1)$-st level U_{k-1}^{2m} in the graph $G_{k,2m}$. The following proposition holds:

LEMMA 2.6: *In any computation on the graph $G_{k,2m}$ using k-pebbles for pebbling the vertex x_i (for $0 \leqq i \leqq 2m-1$) there have to be an open path from an input vertex to x_{i+1} leading through $\alpha_{(i+1)\,\mathrm{mod}\,2}$ at some configuration in which x_i is pebbled.*

Proof: Case $i = 0$: Consider an arbitrary computation S_0 on $G_{k,2m}$ with k pebbles starting with the empty configuration and ending with a configuration in which a pebble is placed on x_0 for the first time. From Lemma 2.4 it follows that k pebbles are needed to close all paths from inputs to the vertex u in the pyramid P_k. Furthermore, if there is at least one path opened in the pyramid P_k, then k pebbles are needed to close P_k. Two important facts follow.

Firstly, a computation on the subgraph $G_{k-1,2m}$ of the graph $G_{k,2m}$ can be started after closing the pyramid P_k with the root u because at the moment of closing the pyramid the subgraph $G_{k-1,2m}$ is empty.

Secondly, the pyramid P_k with the root u have to be maintained as closed during the rest of the computation S_0 because otherwise the computation S_0 would be repeated from some configuration before closing P_k. At least 1 pebble has to be permanently placed on P_k in order to maintain the pyramid P_k closed. Thus all path to x_0 through u are closed. Hence, there have to be a configuration (say C) in the computation S_0 such that 1 pebble is placed on the pyramid P_k with the root u and $k - 1$ pebbles are placed on the pyramid P_{k-1} with the root α_0. That means that in the configuration C there is no pebble on a path leading to the vertex α_1 in $G_{k-1,\,2m}$ and thus there is an open path to x_{i+1} through α_1.

Case $0 < i < 2m - 1$: It is sufficient to repeat arguments used in case $i = 0$ with the addition that permanent pebble keeping closeness of P_k lay on x_{i-1} and thus it maintains all path to x_i through x_{i-1} closed. So Lemma 2.6 is proved. □

THEOREM 2.7: *Let $S(n)$ be a space in the range $\omega(1)$ to $O(n^{1/3})$. Then there is a class of dags $\{G_n\}$, with n vertices and with $S(n)$ pebbles necessary and sufficient to pebble G_n, such that for the time $T(n)$ needed to peeble G_n with $S(n)$ pebbles it holds*

$$T(n) = 2^{\Omega(S(n)\cdot\log n/S(n))}.$$

Proof: Consider the graph $G_{k,\,2m}$ with $k = S(n)$ and $m = c_1 \cdot n/(S(n))$ for appropriately chosen constant $c_1 > 0$. We show that the following recurrent relation (2.1) holds

$$T(G_{k,\,2m}) \geqq b_1 \,.\, m \,.\, T(G_{k-1,\,2m})$$
$$T(G_{1,\,2m}) \geqq b_2 \,.\, m$$

It follows directly that (2.1) holds for $i = 1$. Suppose that (2.1) holds also for $i = 2, 3, \ldots, k - 1$. Following Lemma 2.6 the time needed for pebbling $G_{k,\,2m}$ with $k = S(n)$ pebbles is lower estimated by

$$T(G_{k;\,2m}) \geqq d_2 \,.\, k^2 + \sum_{i=0}^{2m-1} \left[\frac{d_1}{2} T(G_{k-1,\,2m})\right]$$

because for pebbling the pyramid P_k it is needed $1/2\,k\,(k+1)$ computational steps and for pebbling $2m$ vertices $x_0, x_1, \ldots, x_{2m-1}$ it is needed $1/2\,d_1 \,.\, T(G_{k-1,\,2m})$ computational steps. Choose $k = S(n) = c_2 \,.\, n^{1/3}$

for appropriate constant $c_2 > 0$. So we have $m = c_3 . n^{2/3}$ and thus for appropriately chosen constant $b_1 > 0$ it is obtained

$$T(G_{k,2m}) \geqq b_1 . m . T(G_{k-1,2m}).$$

Solving (2.1) one can obtain $T(G_{k,2m}) \geqq d . m^k$ for some constant $d > 0$ and thus for $G_n \equiv G_{S(n), n/S(n)}$ it holds

$$T(n) = 2^{\Omega(S(n) \log n/S(n))}. \quad \square$$

If the minimum space necessary to pebble a n-vertex dag is a constant, then minimum space and polynomial time are achieved simultaneously and the polynomial time is asymptotically tightly expressed in the following form.

CONSEQUENCE 2.8: *Let $T(G_{k,m})$ be the time needed to pebble $G_{k,m}$ with the minimum number of pebbles $S(G_{k,m}) = k$ and let k be a constant. Then*

$$T(G_{k,m}) = \Theta(m^k).$$

Finally we show that the class of dags requiring exponential time for the minimal space standard pebbling is not "resistant" according to the "edge deleting".

PROPOSITION 2.9: *Let $S(n)$ be a space in the range $\omega(1)$ to $O(n^{1/3})$. Then there is a class of dags $\{\hat{G}_n\}$ with n vertices constructed from $\{G_n\}$ by removing one edge and pebbled using the minimal number of pebbles $S(n)$ in time*

$$T(n) = O(n^2).$$

In the graph $G_n \equiv G_{k,m}$ it is sufficient to remove the edge outgoing from the output vertex of the component R_m at the level U_{k-2}^m in order to obtain the required proposition.

3. CONCLUSIONS

The class of dags $\{G_n\}$ of size n pebbleable in space $O(n^{1/3})$ is considered. An asymptotically optimal time complexity for the minimal space standard pebbling is presented. Moreover, it is proved that dags requiring exponential time for the minimal space standard pebbling are not tolerant to the "edge deleting".

An interesting problem might to be find a class of dags of size n which

a. can be pebbled in space $\omega(n^{1/3})$;

b. requires exponential time for the minimal space standard pebbling;

c. is not tolerant to the "edge deleting".

A possible attempt to solve this problem is to construct a class of rooted dags $\{D_k\}$ requiring polynomial time in the minimal space k and having size $o(k^2)$.

Another fruitful problem would be to characterize extreme changes in the time complexity with respect to the "edge deleting" in dags for other variants of the pebble game.

REFERENCES

1. S. A. Cook, An Observation on Time-Storage Trade Off, *Journal of Computers and System Sciences*, 1974, *9*, 308-316.
2. N. Pippenger, Pebbling, *5th IBM Symposium on Mathematical Foundations of Computer Science*, Tokyo, 1980.
3. H. Venkateswaran and M. Tompa, A New Pebble Game that Characterizes Parallel Complexity Classes, *S.I.A.M. J. Computing*, 1989, *18*, 533-549.
4. R. Wilber, White Pebbles Help, *Journal of Computers and System Sciences*, 1988, *36*, 108-124.