1 Upper bound – exploration algorithm with $\mathcal{O}(\log\log n)$ pebbles and $\mathcal{O}(\log\log n)$ bits of memory

Definition. *(s,p,m)-pebble machine is a tuple $(Q,F,P,m,\delta,\delta_{in},\delta_{out},q_0)$, where Q is the set of states, $F\subseteq Q$ is the set of final states, q_0 is the starting state and $\delta\colon (Q\setminus F)\times\{0,1\}\to\{0,1\}\times Q\times\{L,R\}$ is a Turing machine M. Moreover M is allowed to observe the local environment at the current vertex by*

$$\delta_{in}\colon Q\times 2^P\times 2^P\times\{0,1,\dots\Delta-1\}^2\to Q,$$

where the input is the current state, the set of pebbles carried, the set of pebbles placed in the current vertex, the degree of each vertex and label of the edge agent used to come to this vertex and M is allowed to perform actions determined by function

$$\delta_{out}\colon Q\to 2^P\times 2^P\times\{0,1,\dots,\Delta-1\},$$

where the output says which pebbles are picked up (if they are in the current vertex), which pebbles are placed on the current vertex (if the agent carries them) and which label of edge is used to move agent in the graph.

Definition. *A traversal sequence is a sequence of integers $\ell_0,\ell_1,\dots$ that guides agent along the graph such that in i-th step the agent traverses an edge labeled ℓ_i.*

Definition. *An exploration sequence is a sequence of integers $e_0,e_1,\dots$ that guides agent along the graph as follows: Agent A starts in vertex v_0 with $l_0=0$ and in each step i he moves from vertex v_{i-1} to a vertex v_i by traversing the edge labeled l_i+e_i from v_i, where l_i is the label of the edge leading back to the previous position of agent A.*

Definition. *An exploration (or traversal) sequence e is universal for class of graphs $\mathcal{G}$ iff for all graphs $G\in\mathcal{G}$ and any starting vertex v the agent A following the sequence e traverses whole graph G.*

Theorem (3.1, Reingold)**.** *There is an $\mathcal{O}(\log n)$-space algorithm producing a universal exploration sequence for any regular graph on n vertices.*

Theorem (3.2)**.** *There exists a $(\mathcal{O}(1),0,\mathcal{O}(\log n))$-pebble machine that moves along a closed walk and either explores the graph or visits at least n distinct vertices, for any graph with bounded degree.*

Auxiliary lemmas:

Lemma (3.1)**.** *An agent following an exploration sequence of the form $(e_0,\dots,e_{k-1})^*$ in an undirected graph moves along a closed walk.*

Lemma (3.2)**.** *There exists an $\mathcal{O}(\log n)$-space algorithm producing a universal exploration sequence $(e_0,\dots,e_{k-1})*$ for any 3-regular graph with at most n vertices.*

Theorem (3.3)**.** *There is a constant $c\in\mathbb{N}$, such that for any graph G with bounded degree and any $(s,p,2m)$-pebble machine M, there exists a $(cs,p+c,m)$-pebble machine M' that simulates the walk of M or explores G.*

Auxiliary claims in the proof:

1. We can find a closed walk ω with 2^{m_1} distinct vertices such that pebbles placed to this walk can encode the memory of M.
2. We can move along ω while counting how many distinct vertices we seen while this number is at most 2^{m_1}.
3. We can read from and write to this memory.
4. We can move this memory to new closed walk ω' starting with next vertex v' (v' is a neighbor of v where walk ω starts).

Theorem (3.4 + Corollary 3.1)**.** *Any bounded-degree graph on at most n vertices can be explored using $\mathcal{O}(\log\log n)$ pebbles and memory.*

More over this exploration algorithm can be adapted to eventually return to the starting vertex and terminate after $n^{\mathcal{O}(\log\log n)}$ steps, without additional memory or pebbles, even if the number of vertices of the graph is not known a priori.

2 Lower bound

Definition. *For $r \leq p$ the graph B is an r-barrier for an agent A with ppebbles if, in the above setting, the following holds for all graphs G and every pair (a, b) in $\{v, v'\} \times \{u, u'\}$: If starts in an arbitrary vertex in G, then it never traverses B from a to b or vice versa with a set of at most r pebbles carried by the agent, observed or placed on vertices of B during the traversal. We equivalently say that A cannot traverse B from a to b or vice versa while using at most r pebbles.*

Lemma (4.1)**.** *Given a p-barrier with m vertices, we can construct a trap with $2m + 4$ vertices.*

Theorem (4.1, Fragniaud et al.)**.** *For any q non-cooperative s-state agents without pebbles, there exists a 3-regular graph G on $\mathcal{O}(qs)$ vertices with the following property: There are two edges $\{v_1, v_2\}$ and $\{v_3, v_4\}$ in G, the first labeled 0, such that any of the q agents starting in v_1 or v_2 does not traverse the edge $\{v_3, v_4\}$.*

Lemma (4.2)**.** *For every s-state agent A with p pebbles there exists a 0-barrier with $\mathcal{O}(s^2)$ vertices, which is independent of the starting state of A.*

Definition (4.2)**.** *The configuration of an s-state agent A with p pebbles in a vertex v is a $(p + 1)$-tuple $(q, P_1, P_2, \ldots, P_p)$, where q is the current state of A and P_i is the position of the i-th pebble relative to the agent (lexicographically smallest shortest traversal sequence or (-1) if p is carried by A).*

Lemma (4.3)**.** *Let B be a $(p - 1)$ barrier for an agent A with p pebbles. Then, the following hold for any graph of the form $G(B)$:*

1. *A cannot get from a macro vertex v to a distinct macro vertex v' while using less than p pebbles.*
2. *At any time, there is some macro vertex v such that A and each pebble are at v or in one of the surrounding gadgets $B(0), B(1), B(2)$.*

Lemma (4.4)**.** *Let B be a $(p - 1)$-barrier for an agent A, G, G' be two 3-regular graphs, $v_0, v_1, \ldots$ the sequence of macro vertices visited by A in $G(B)$ and $\ell_0, \ell_1, \ldots$ be the corresponding macro traversal sequence. Similarly, let $v'_0, v'_1, \ldots$ the sequence of macro vertices visited by A in $G'(B)$ and $\ell'_0, \ell'_1, \ldots$ be the corresponding macro traversal sequence. If at some point the configuration of A in v_i is the same as the configuration of A in v'_j, then $\ell_{i+k} = \ell'_{j+k}$ holds for all $k \in \mathbb{N}$.*

Theorem (4.2)**.** *Given an $(r - 1)$-barrier B with m vertices for an agent A with $p \geq r$ pebbles, we can construct an r-barrier B' with $\mathcal{O}(\binom{p}{r}\alpha_B^2)$ vertices for A.*

Lemma (4.5)**.** *The maximum number of possible configurations α_r of $A_r^{(k)}$ in a macro vertex in a graph of the form $G(B_{r-1})$ can be bounded by $sr! \prod_{i=1}^{r} (2^{3i} m_{r-i} + 2^{3i+1})$.*

Theorem (4.3)**.** *For $r \leq p$ and $s \geq 2^p$, the number of vertices of the r-barrier B_r for the s-state agent A with p pebbles is bounded by $\mathcal{O}(s^{8^{r+1}})$.*

Theorem (4.4)**.** *For any s-state agent with p pebbles with $s \geq 2^p$ there exists a trap with at most $\mathcal{O}(s^{8^{p+1}})$ vertices.*

Theorem (4.5)**.** *For any constant $\varepsilon > 0$, an agent with at most $\mathcal{O}((\log n)^{1-\varepsilon})$ bits of memory needs at least $\Omega(\log\log n)$ distinguishable pebbles for exploring all graphs on at most n vertices.*