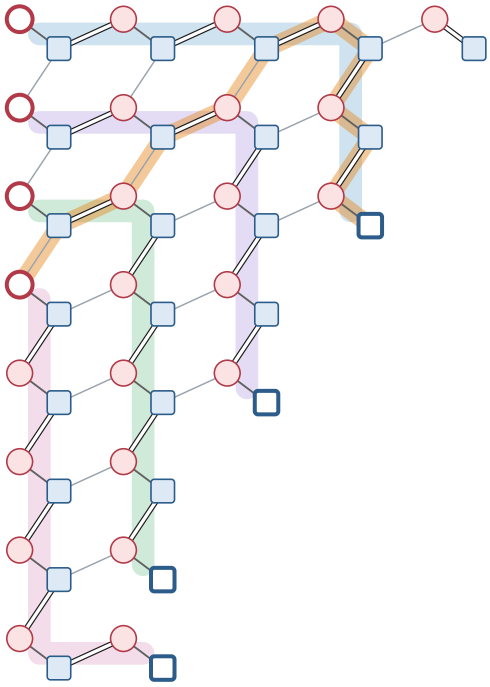
How slow can Algorithm 7.5.1H be?

Rounds, staircases, and Tchoukaillon graphs

Notes on exercises 15 and 16 of
The Art of Computer Programming, Pre-Fascicle 14A,
Section 7.5.1: Bipartite Matching



Claude Opus 5.5

3 November 2026
References are to the pre-fascicle of 18 September 2026 (zeroth printing, revision −88)

Abstract

Knuth's Algorithm 7.5.1H (Algorithm H for short), his implementation of the Hopcroft–Karp algorithm, works in *rounds*: each round builds a dag of shortest augmenting paths and then removes a maximal set of disjoint ones. Exercise 15 of Pre-Fascicle 14A asks how badly this can go, and proposes a candidate worst case, the *Tchoukaillon graph*, whose number of rounds is governed by the Tchoukaillon numbers $\text{Ч}_n \approx n^2/\pi$. These notes answer the unrestricted version of the question, which the answer to exercise 16 poses as harder. A graph whose maximum matching has size s can force at most $\lfloor\sqrt{4s+1}\rfloor - 1$ rounds, and this is attained for every s by explicit *staircase graphs*: planar graphs of maximum degree three, with input orders under which we can prove exactly what Algorithm H does. Equivalently, n rounds are possible exactly when $s \geq \lfloor(n+1)^2/4\rfloor$. The smallest size at which this beats a Tchoukaillon graph is $s = 9$, and the advantage grows like $(2-\sqrt{\pi})\sqrt{s}$. On the same graphs Algorithm H performs $\Theta(E\sqrt{V})$ operations. We also show that the literal definition in exercise 15 can be met only when $s \in \{1, 2, 3, 4, 5, 8, 9\}$; Knuth's own 16-pair example meets a weaker condition, which we make precise. Whether that weaker condition can be met for every s is still open, and we collect what is known about it.

Contents

1 Overview **1**

2 Algorithm H in brief **3**
2.1 Matchings and augmenting paths . . . 3
2.2 Shortest augmenting paths and rounds . . . 4
2.3 What an input is . . . 4
2.4 A worked round . . . 5
2.5 Counting rounds and operations . . . 7

3 Exercise 15 and Tchoukaillon graphs **7**
3.1 Basic paths and parts (a) and (b) . . . 8
3.2 The schedule and the Tchoukaillon numbers . . . 8
3.3 Knuth's sixteen-pair example . . . 9
3.4 Three readings of the definition . . . 10

4 How many rounds can there be? **11**
4.1 A rectangle that must fit . . . 11
4.2 Proof of the upper bound . . . 12
4.3 What a run with the most rounds must look like . . . 12

5 Staircase graphs **13**
5.1 The graph and its input . . . 13
5.2 Watching the smallest interesting case . . . 14
5.3 Hooks . . . 15
5.4 No shortcuts . . . 17
5.5 The forced search . . . 17
5.6 How much one path blocks . . . 19

5.7 Counting the work 20

6 The true maximum against the Tchoukaillon schedule 20
6.1 Always $T(s) \leq R(s)$ 21
6.2 Where the true maximum is larger 21
6.3 Why nine is the first size 21
6.4 Initial deficiency 22

7 The literal definition 22
7.1 Uniform consecutive levels force halving 22
7.2 Consequences 24

8 The operational question 24
8.1 Small sizes 25
8.2 Operational is weaker than basic-row 25
8.3 The shape forced by the first matching 26
8.4 Realizations need cycles 26
8.5 The input order matters 28
8.6 What an induction would need 30

9 Simpler examples for the running time 30

10 Credits, literature, and open problems 32
10.1 Where the ideas come from 32
10.2 Open problems 32

A An input order for Knuth's sixteen-pair graph 33

B The halving graphs 34

C Computed realizations for small sizes 35

D Programs and checks 35

1 Overview

Algorithm 7.5.1H, in Knuth's Pre-Fascicle 14A [11], is his implementation of the Hopcroft–Karp algorithm [8]. Inside Section 7.5.1 Knuth calls it simply Algorithm H, and so shall we. After a greedy start (step H1) it repeats a *round*: build a dag that contains every shortest augmenting path, then dismantle the dag, augmenting along a maximal collection of vertex-disjoint shortest paths. Corollary K bounds the number of rounds by about $2\sqrt{s}$, where s is the size of a maximum matching, so the running time is $O\big((M+N+E)\sqrt{s}\,\big)$.

Is that square root ever needed? Knuth remarks on page 5 that graphs needing anywhere near $\sqrt{s}$ rounds are hard to find, and that even for such a graph "its edges have to appear in a particularly perverse order." Exercise 15 proposes a candidate: a *Tchoukaillon graph*, on which each round removes as few shortest paths as a simple counting argument allows. The resulting round count is governed by the Tchoukaillon numbers Ч_n of exercise 11, and since $\text{Ч}_n \approx n^2/\pi$, such a graph would force about $\sqrt{\pi s}$ rounds. Exercise 16 asks whether Tchoukaillon graphs exist for every s, and its answer raises an even harder question: can some graph force *more* rounds than a Tchoukaillon graph would? Fig. 1 shows the answer.

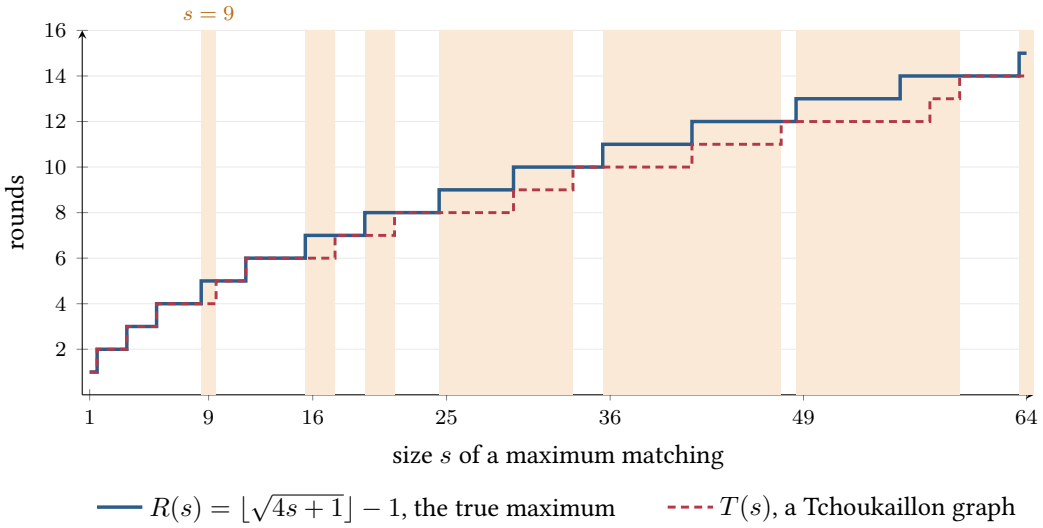


Figure 1. The largest possible number of rounds $R(s)$ (blue), against the number of rounds $T(s)$ that a Tchoukaillon graph would need (red), for matching sizes $s \le 64$. Shaded columns mark the sizes where $R(s) > T(s)$; the first is $s = 9$. The two staircases step up at the quarter-squares Q_n and at the Tchoukaillon numbers Ч_n (both defined below).

The answers in brief

1. **The maximum number of rounds.** Count H1 as the first round, as answer 15(c) does. A bipartite graph whose maximum matching has s edges can make Algorithm H perform at most

$$R(s) = \lfloor\sqrt{4s+1}\rfloor - 1$$

rounds, whatever the order of its input, and for every s some graph and input order attain this bound (Theorem R). Put differently, n rounds are possible exactly when

$$s \;\ge\; Q_n = \left\lfloor \frac{(n+1)^2}{4} \right\rfloor \qquad \text{(the quarter-squares } 1, 2, 4, 6, 9, 12, 16, 20, \ldots\text{)},$$

while a hypothetical Tchoukaillon graph has at least n rounds exactly when $s \geq \text{Ч}_n$ $(1, 2, 4, 6, 10, 12, 18, 22, \ldots)$.

2. **Staircase graphs.** The extremal graphs are explicit, planar, and have maximum degree three. Their input orders make Algorithm H find exactly one shortest augmenting path per round, at levels $L = 2, 3, \ldots, n$, and we can prove every step of the execution (Theorem S). Algorithm H performs $\Theta(E\sqrt{V})$ elementary operations on them (Theorem W), so Corollary K is tight even for sparse planar graphs.
3. **Beating Tchoukaillon.** Always $Q_n \leq \text{Ч}_n$, so the true worst case is always at least as slow as a Tchoukaillon graph would be. The first size where it is strictly slower is $s = 9$: five rounds are possible, against four. Asymptotically $R(s) - T(s) \sim (2 - \sqrt{\pi})\sqrt{s}$ (Section 6).
4. **The literal definition.** Exercise 15(c) asks that *every* augmenting path have the current shortest level at every round. Graphs meeting that requirement exist only when $s \in \{1, 2, 3, 4, 5, 8, 9\}$ (Corollary L), because consecutive rounds of this kind must at least halve the number of free girls (Theorem U). Knuth's 16-pair example in answer 15(d) has augmenting paths of levels 2 and 3 immediately after H1, yet with suitable input orders it does follow the Tchoukaillon schedule.
5. **The open question.** Can the Tchoukaillon *schedule* (the numbers of free girls and of paths removed, round by round) be realized for every s? Explicit inputs exist for all $s \leq 47$, and they even keep f disjoint SAPs in every round, as Knuth's matrices do. Any family of inputs that realizes the schedule for unboundedly large s must contain at least $s/12 - O(1)$ independent cycles (Theorem C). Some candidate graphs fail only because of the order in which Algorithm H searches: its queue, the head insertion of step H6, and the replacement rule of step H16 (Section 8.5).

How these notes are organized

Section 2 recalls just enough of Algorithm H to make precise what an "input" is and which choices it leaves open; it also works through one round of an example that recurs throughout. Section 3 restates exercise 15, explains where the Tchoukaillon numbers come from, and separates three ways of reading the definition of a Tchoukaillon graph. Sections 4 and 5 prove Theorem R: first the upper bound, which is a sharpened form of the counting in Theorem H, then the staircase construction. Section 6 compares the answer with the Tchoukaillon numbers. Section 7 settles the literal reading of exercise 15, and Section 8 collects what we know about the open, numerical reading. Section 9 gives simpler examples for the running-time question alone. Section 10 discusses the literature.

The appendices give a complete ordered input for Knuth's 16-pair graph, details of a family that meets the literal definition, a catalog of computed examples, and a guide to the accompanying programs.

Which draft. Page, exercise, step, and equation numbers refer to the zeroth printing of Pre-Fascicle 14A, revision −88, dated 18 September 2026. A pre-fascicle is a working draft, so these numbers may change in later revisions.

What is classical and what is new

The tools are the ones in the pre-fascicle: Berge's augmenting paths [1], the layered dag and the fact that the shortest level increases from round to round [8], and the counting argument behind Theorem H. The upper bound in Theorem R refines Hopcroft and Karp's bound on the number of

phases. As far as we know, the staircase graphs, the exact value of $R(s)$, and the results of Sections 7 and 8, are new; Section 10 discusses the literature.

2 Algorithm H in brief

This section recalls the parts of Section 7.5.1 that matter for worst cases. Readers who know pages 1–5 of the pre-fascicle well can skim it, but should look at two things: the definition of an *input* in Section 2.3, which fixes exactly what freedom a worst-case construction has, and the worked round in Section 2.4.

2.1 Matchings and augmenting paths

A bipartite graph has M *girls* and N *boys*, joined by E edges; each edge joins a girl to a boy. A *matching* is a set of edges with no common vertex. Its edges are called *couples*, and a vertex that belongs to no couple is *free*. A matching is *maximal* if no edge can be added to it, and *maximum* if no matching has more edges. Throughout, s denotes the size of a maximum matching. We often write $V = M + N$ for the number of vertices.

We follow the pre-fascicle's drawing conventions. When two matchings are compared, an edge of the reference matching is written x — y and an edge of the current matching M' is written x ═ y. In particular an *augmenting path* for M' is a path

$$g_0 \text{ — } b_1 \text{ ═ } g_1 \text{ — } b_2 \text{ ═ } \cdots \text{ ═ } g_{L-1} \text{ — } b_L \tag{2.1}$$

that starts at a free girl, ends at a free boy, and alternates between edges outside and inside M'. It has $2L - 1$ edges, and we say it has *level* L. Exchanging the two kinds of edges along the path (Fig. 2) produces a matching with one more couple. In (2.1) no reference matching is involved, and the single lines mean "an edge not in M'." We describe augmenting paths by their level; when the number of edges matters we say so. An edge named without regard to any matching is written by juxtaposition, as in $g_1 b_6$.

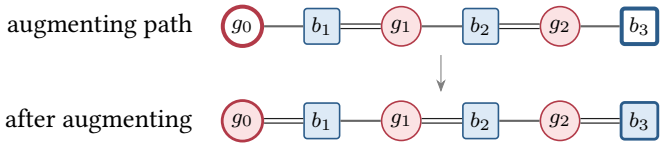


Figure 2. An augmenting path of level $L = 3$, and the matching after augmenting along it. Double lines are couples; heavy outlines mark free vertices. In all figures girls are drawn as circles and boys as squares.

Berge's observation [1] (pre-fascicle, page 1) is that if M' has s' couples, then the union of M' with a maximum matching contains at least $s - s'$ *vertex-disjoint* augmenting paths for M'. We shall use this mostly when $s = M = N$, so that there is a *perfect* matching P. Then the symmetric difference $M' \mathbin{\triangle} P$ consists of exactly f vertex-disjoint augmenting paths, where $f = M - s'$ is the number of free girls, together with some alternating cycles. Following answer 15(a), we call these f paths the *basic paths* of M' with respect to P. A basic path of level ℓ contains exactly ℓ edges of P, and like Knuth in answer 15(d) we often draw it as a *row* of ℓ cells, one for each girl–boy pair of P that it visits. (Later sections call these rows the rows of a matrix, or hooks; they are always basic paths.)

2.2 Shortest augmenting paths and rounds

A *shortest augmenting path*, or SAP, is an augmenting path of smallest level. To find them, orient every edge relative to the current matching: an edge outside the matching points from its girl to its boy, and a couple points from its boy to its girl. Alternating paths that start at a free girl are then exactly the directed paths that start there. The *distance* of a vertex is the number of edges on a shortest such directed path from some free girl to it.

Hopcroft and Karp [8] organize all SAPs into a directed acyclic graph, built breadth-first from the free girls (page 2 of the pre-fascicle). The dag has *levels* $G_0, B_1, G_1, \ldots, G_{L-1}, B_L$: G_0 is the set of free girls; B_k consists of the boys first reached from girls in G_{k-1}; and G_k consists of the mates of the boys in B_k. So G_k and B_k are the girls at distance $2k$ and the boys at distance $2k-1$. There is an arc $b \to g$ for each edge $g \text{ — } b$ with $g \in G_{k-1}$, $b \in B_k$, and an arc from each girl to her mate. The construction stops at the first level B_L that contains a free boy. Knuth adds two special nodes: arcs $g \to \bot$ from the free girls and $\top \to b$ to the free boys of B_L. The SAPs are then exactly the paths from $\top$ to $\bot$, read backwards; this is why the arcs in the pre-fascicle's figure (7) point to the left.

Each *round* of Algorithm H builds this dag (steps H2–H10) and then dismantles it (steps H11–H17): it repeatedly finds a path from $\top$ to $\bot$ by depth-first search, augments the matching along it, and deletes its vertices from the dag. When no path remains, the paths removed form a *maximal* set of vertex-disjoint SAPs: no further SAP is disjoint from all of them. This set need not be as large as possible, and the gap between "maximal" and "maximum" is what a worst-case input exploits.

After a round at level L, every augmenting path has level greater than L; this is exercise 5, and goes back to Hopcroft and Karp [8, Theorem 2 and Corollaries 3–4]. So the levels of successive rounds strictly increase. Theorem H turns this into a counting bound: when a round at level L is over, every one of the $s - s'$ disjoint augmenting paths has at least $L+1$ edges of a maximum matching, so $s \geq (L+1)(s-s')$. Corollary K deduces that there are at most about $2\sqrt{s}$ rounds, each costing $O(M+N+E)$.

2.3 What an input is

Algorithm H is deterministic. To speak of a graph that "might cause" it to behave badly, as exercise 15(c) does, we must say exactly which features of the input it reads. There are two:

(i) the numbering of the girls $1, \ldots, M$, which fixes the order in which step H1 scans them;

(ii) for each girl g, the order of her list of boys, which starts at `GLINK`$[g]$ and continues through `GNEXT`.

We call a bipartite graph together with these data an *ordered graph*. By exercise 6, any desired list order arises from a suitable order of the input pairs, since each new edge is inserted at the front of its girl's list. Nothing else matters. The numbering of the boys is irrelevant, since no step compares boy indices. There is also *no* boy-side order to choose: the reverse lists `BLINK`$[b]$ are built by the algorithm itself.

For orientation, here is what the steps do, in our words (the precise code is on pages 4–5 of the pre-fascicle).

H1	Greedy start: a maximal matching; the free girls go into QUEUE.
H2–H3	Start a dag; start the next level.
H4–H7	Take the next girl of the current level and scan her list of boys.
H8–H9	A boy seen for the first time joins the next level; his mate is queued for the level after, or, if he is free, the final level L is fixed and he is recorded as a root.
H10	Go on to the next level, or finish the dag (or stop if no free boy was reached).
H11	Take the next root; if none is left, reset all marks and start a new round.
H12–H15	Depth-first search from the root down the reverse lists BLINK, consuming each arc it follows and skipping used girls; back up when a list runs out.
H16–H17	A free girl was reached: remove her from the free list and augment along the path.

The following consequences of the code will be used repeatedly.

- **H1 scans the girls in decreasing order.** Answer 7 matches girl g to the first free boy on her list, for $g = M, M-1, \ldots, 1$, and appends each unsuccessful girl to QUEUE. So the free girls start out in QUEUE in decreasing numerical order. The resulting matching is maximal, so the first dag has $L \geq 2$.
- **The dag levels are scanned in queue order.** Girls of G_l are processed in their order in QUEUE; each scans her own list in order. A boy enters B_{l+1} when he is first seen (step H8), and his mate is appended to QUEUE as a girl of G_{l+1} (step H9). So the order of G_{l+1} is the order in which the boys of B_{l+1} were discovered.
- **Reverse lists are built by insertion at the head.** The first girl who sees b becomes the only entry of BLINK$[b]$ (step H8); every later girl of the same level who is adjacent to b is inserted at the *front* (step H6). Hence BLINK$[b]$ lists the girls of G_l adjacent to b in the *reverse* of their order in QUEUE.
- **The last level is cut short.** When the first free boy is found, L is set to the current level and the queue of the next level is discarded. After that, matched boys are no longer entered (step H8), but free boys still are. The free boys of B_L are the last r entries of MARKED, in order of discovery, and step H11 uses them as roots of the depth-first search in that order.
- **The depth-first search destroys what it reads.** Step H13 removes each arc from BLINK$[b]$ as it follows it, and skips a girl if she has already been used by an earlier SAP of this round (she is "deleted"). If a list becomes empty, step H15 backs up one level; nothing is ever re-examined. Reaching a free girl completes a SAP.
- **A used free girl is replaced by the last one.** Step H16 removes the girl g_0 of a new SAP from the free list by moving the last free girl, QUEUE$[f-1]$, into her slot. So the order of the free girls is preserved only if each SAP ends at the last of them.

2.4 A worked round

The ordered graph of Fig. 3 has nine girls and nine boys. Its girl lists, in increasing order of girls, are

$$[6,8];\ [4,6];\ [1,4];\ [1];\ [1,2];\ [2,3];\ [4,2,5];\ [6,5,7];\ [8,9]. \tag{2.2}$$

(Girl 1 is adjacent to boys 6 and 8, in that order, and so on.) Its vertices are arranged in nine *cells*: cell (a,c), in row a and column c, holds one girl and one boy, joined by an edge (the short dark edge inside the cell). These nine edges form a perfect matching, so $s = 9$. The other edges join the boy of a cell either to the girl of the next cell in the same row or to the girl of the cell below; in the drawing they run diagonally, up-right and down-left respectively. We shall meet this graph again in Section 5 as the staircase graph Γ_5.

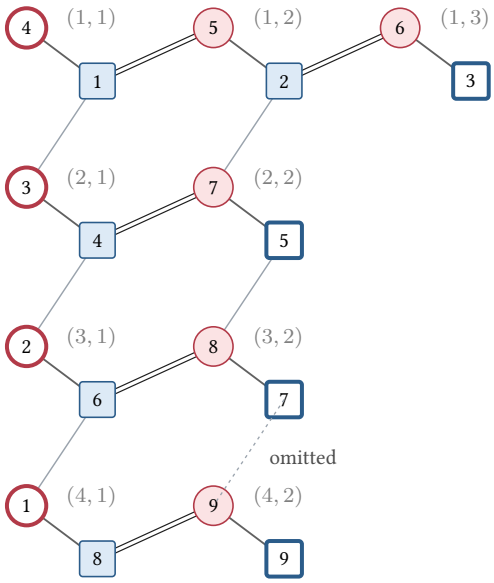


Figure 3. The 9-pair example, with the matching produced by H1. The grey label (a, c) names the cell of the girl beside it and of the boy below her; the dark edge joining them belongs to a perfect matching. Light grey lines are the other edges of the graph, double lines are couples, and free vertices have heavy outlines. The dotted segment marks an edge deliberately left out of the graph (Section 5.1).

Step H1. Girls $9, 8, 7, 6, 5$ come first. Each takes the first boy on her list, namely $8, 6, 4, 2, 1$, and these are all different. Girls $4, 3, 2, 1$ then find every boy on their lists taken. So H1 produces the five couples drawn as double lines in Fig. 3, and

$$\texttt{QUEUE} = (4, 3, 2, 1), \qquad f = 4.$$

In the picture, the free girls are the first girls of the four rows, and the free boys are the last boys of the rows.

Building the dag. Fig. 4 shows the dag of the first round. Girl 4 scans her list $[1]$ and discovers boy 1. Girl 3 scans $[1, 4]$: boy 1 has already been seen at this level, so girl 3 is inserted at the head of $\texttt{BLINK}[1]$, giving $\texttt{BLINK}[1] = (3, 4)$; then she discovers boy 4. Girls 2 and 1 behave the same way, so $B_1 = \{1, 4, 6, 8\}$ in that order, and their mates $5, 7, 8, 9$ form G_1. At the next level boy 2 is discovered first; he is matched (to girl 6), so the construction goes on. Then girl 7 discovers boy 5, who is free: now $L = 2$. Boys 7 and 9 are free too, so the roots of the search will be $5, 7, 9$, in that order.

Dismantling the dag. The search starts at boy 5. The head of his list is girl 8, whose mate is boy 6; the head of $\texttt{BLINK}[6]$ is girl 1, who is free. That completes the SAP

$$g_1 \text{ — } b_6 \text{ = } g_8 \text{ — } b_5,$$

and Algorithm H augments along it. The next root, boy 7, has only girl 8 on his list, and she has just been used. The last root, boy 9, leads to girl 9 and then to boy 8, whose only remaining entry is girl 1, also used. So the round ends with a *single* SAP, even though the four rows of Fig. 3 contain three disjoint SAPs (rows 2, 3, and 4). The one path that was found meets all three of them. Step H16 removed girl 1, the last free girl, so the queue for the next round is $(4, 3, 2)$.

The remaining rounds behave similarly: each finds one SAP, at levels 3, 4, and 5, so the whole run has five rounds counting H1. Fig. 9 on page 15 shows all of them. A Tchoukaillon graph with $s = 9$ would need only four.

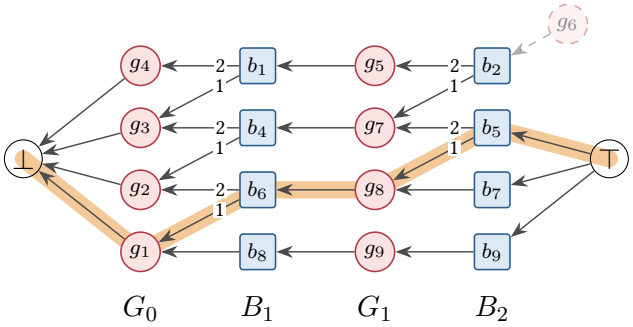


Figure 4. The first dag for the graph of Fig. 3, drawn as in the pre-fascicle's figure (7). When a boy has two arcs, the small numbers give their order in `BLINK`; the girl who discovered him comes *second*. The shaded path is the only SAP that Algorithm H finds. Boy 2 belongs to B_2 but is matched, so his mate (faded) is not part of the dag.

2.5 Counting rounds and operations

Answer 15(c) gives the number of rounds of a Tchoukaillon graph as "the final value of L." In such a graph the dags have levels $2, 3, \ldots, L$, so this count treats step H1 as a round at level 1; answer 7 makes the same identification when it notes that, if H1 did not produce a maximal matching, the first round would do so with $L = 1$. We adopt this convention.

Definition 2.1. A round is *successful* if it enlarges the matching. The *number of rounds* of a run of Algorithm H is the number of successful rounds, where step H1 counts as one round (successful whenever the graph has an edge). The final dag construction, which finds no augmenting path and stops the algorithm, is not counted.

Thus a run on a graph with $s \geq 1$ has $1 + k$ rounds if k dags contain augmenting paths. If the levels of those dags are $2, 3, \ldots, n$ then there are exactly n rounds. The run of Section 2.4 has five.

For running times we count elementary operations of Algorithm H rather than seconds. In each successful round we record three counts, named after the steps that perform them:

$$W_5 = \text{number of adjacency entries examined in step H5 (the breadth-first search),}$$
$$W_{13} = \text{number of arcs consumed in step H13 (the depth-first search),}$$
$$W_{11} = \text{number of marks reset in step H11, which equals the number of boys in the dag.}$$

W_{13} includes arcs that lead to deleted girls. Every other step of a round costs $O(1)$ per unit of one of these counts, plus $O(f)$, so the total work in rounds after H1 is $\Theta(W_5 + W_{13} + W_{11})$ up to that small additive term. For example, the round of Section 2.4 has $W_5 = 17$ (girls $4, 3, 2, 1, 5, 7, 8, 9$ have lists of total length 17), $W_{11} = 8$ (the boys of B_1 and B_2), and $W_{13} = 5$ (two arcs for the successful root, one for boy 7, and two for boy 9).

3 Exercise 15 and Tchoukaillon graphs

Exercise 15 assumes that the input has a perfect matching, $s = M = N$, and looks for a graph on which L increases by only 1 per round while each round removes as few SAPs as possible. This section explains the counting behind the exercise, where the Tchoukaillon numbers enter, and why the definition of a Tchoukaillon graph can be read in more than one way.

3.1 Basic paths and parts (a) and (b)

Fix a perfect matching P and a current matching with f free girls, whose next dag has level L. Recall that the basic paths are the f vertex-disjoint augmenting paths in the symmetric difference of the current matching with P. We also need the number

$$\nu = \text{the largest number of vertex-disjoint SAPs of the current matching.}$$

Always $\nu \le f$, since each SAP uses its own free girl.

Lemma 3.1 (Exercise 15(a), (b))**.**

(a) If every augmenting path of the current matching has level L, then the dag satisfies $f \le \min\{|G_0|, |B_1|, \ldots, |G_{L-1}|, |B_L|\}$.

(b) Every maximal set of vertex-disjoint SAPs has at least $\lceil \nu/(L+1) \rceil$ members. In particular, if there are f vertex-disjoint SAPs (as there are under the hypothesis of (a)), steps H11–H17 *find at least $\lceil f/(L+1) \rceil$ of them.*

Proof. (a) Under the hypothesis, the f basic paths are SAPs. Let $g_0 - b_1 = g_1 - \cdots - b_L$ be one of them. Its first part shows that g_j has distance at most $2j$. If the distance were smaller, a shortest directed path to g_j followed by the rest of the SAP would be an alternating walk from a free girl to a free boy with fewer than $2L-1$ edges; cutting out the closed loops of that walk would leave an augmenting path of smaller level, which is impossible. So $g_j \in G_j$, and likewise $b_j \in B_j$. Disjoint paths contribute different vertices to each level.

(b) Let S be a SAP. Its interior vertices are matched, and any augmenting path through a matched vertex also contains that vertex's mate. So a path that meets S meets it at g_0, at b_L, or in one of the $L-1$ couples of S. Paths of a vertex-disjoint family can share none of these, so S meets at most $L+1$ members of the family. Take a family of ν disjoint SAPs: each of them must meet some path of the maximal set (otherwise it could be added), and the bound follows. □

The hypothesis of (a) is stronger than what (b) needs. This difference between "all augmenting paths have level L" and "there are f disjoint SAPs" leads to the three readings of Section 3.4, and the gap between ν and f is what the worst cases of Section 5 exploit.

3.2 The schedule and the Tchoukaillon numbers

A maximal matching has at least $\lceil s/2 \rceil$ couples, because each of the s edges of P must touch one of its couples and each couple touches at most two of them. So H1 leaves at most $\lfloor s/2 \rfloor$ girls free. Exercise 15(c) asks for a graph that starts with this largest possible number of free girls and, at every level $L = 2, 3, \ldots$, removes only the number of SAPs that Lemma 3.1(b) forces when $\nu = f$. The numbers of free girls and of SAPs removed then follow the *Tchoukaillon schedule*

$$f_2 = \lfloor s/2 \rfloor, \qquad a_L = \lceil f_L/(L+1) \rceil, \qquad f_{L+1} = f_L - a_L = \lfloor L f_L/(L+1) \rfloor, \tag{3.1}$$

which stops when f reaches 0. For $s = 16$ it reads:

level L	2	3	4	5	6
free girls f_L	8	5	3	2	1
SAPs removed a_L	3	2	1	1	1

With H1 counted as round 1, that is six rounds.

Let $T(s)$ be the number of rounds of the schedule, that is, the last level L with $f_L > 0$ (and $T(1) = 1, T(0) = 0$). Answer 15(c) observes that (3.1) is the recurrence of exercise 11(b) for the solitaire game Tchoukaillon: in Knuth's notation there, $f_L = l_{L-1,s}$ and $a_L = m_{L,s}$, by parts (ii) and (i). Consequently

$$T(s) = n \quad \Longleftrightarrow \quad \text{Ч}_n \le s < \text{Ч}_{n+1}, \tag{3.2}$$

where the *Tchoukaillon number* Ч_n is the least number of stones for which the winning strategy uses pit n:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Ч_n	1	2	4	6	10	12	18	22	30	34	42	48	58	60

One can also see (3.2) directly. Run the schedule backwards: the last round has $f_n = 1$, and the smallest f_L that leads to a given f_{L+1} is $\lceil (L+1)f_{L+1}/L \rceil$. This gives Knuth's formula $\text{Ч}_n = \lceil 2\lceil \frac{3}{2}\lceil \frac{4}{3} \cdots \lceil \frac{n}{n-1} \rceil \cdots \rceil \rceil \rceil$ of exercise 11(c). Exercise 11(d) shows that $\text{Ч}_n = n^2/\pi + O(n^{4/3})$, so $T(s) = \sqrt{\pi s} + O(s^{1/6})$.

These are statements about the *numbers* in (3.1). Whether any graph realizes them is the question of exercise 16.

3.3 Knuth's sixteen-pair example

Answer 15(d) describes a graph with $s = 16$ by a sequence of matrices (Fig. 5). Label the edges of the perfect matching g_k — b_k, $1 \le k \le 16$. A row such as 1 2 4 stands for the basic path g_1 — b_1 ═ g_2 — b_2 ═ g_4 — b_4: its cells are the pairs it visits, and moving one cell to the right means following a couple. Each matrix lists the basic paths before one round. All its rows have L cells, so they are f disjoint SAPs and $\nu = f$. The graph has the 16 edges $g_k b_k$ and 19 more, listed in (A.1) of Appendix A.

An edge $g_i b_j$ with $i \ne j$ jumps from pair i to pair j. Every SAP that Algorithm H finds in this graph moves through the current matrix by following couples to the right along a row and by such jumps from row to row within a column. For example, the first round finds the SAP g_5 — b_3 ═ g_4 — b_2: it starts at the first cell of row 5 6, jumps to the first cell of row 3 4, moves right along that row, and jumps up to the last cell of row 1 2, whose boy is free. Augmenting replaces these three rows by the two rows 1 2 4 and 3 5 6 of the next matrix. In general a SAP that meets k rows leaves $k - 1$ rows, each with one more cell.

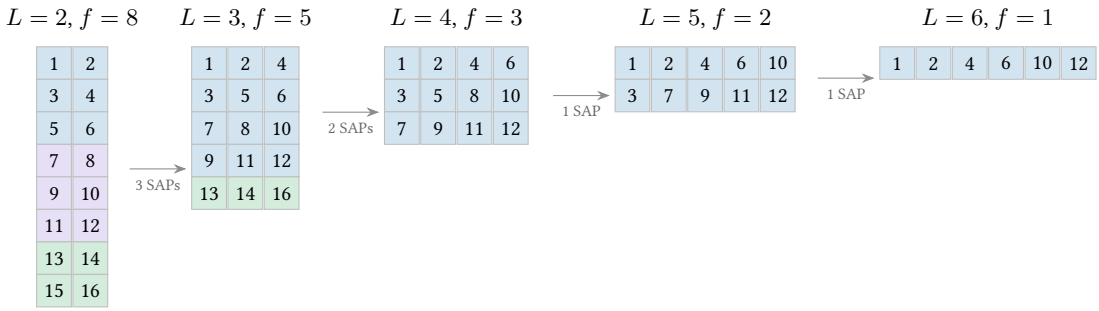


Figure 5. The matrices of answer 15(d). Each row is a basic path, given by the indices of the pairs of the perfect matching that it visits. Within each matrix, rows of the same colour are merged by one SAP of the next round (colours are assigned separately in each matrix); the green row at $L = 3$ is itself a SAP, and augmenting along it matches all three of its pairs by P.

The first matrix says that the matching after H1 consists of the couples $b_1 g_2$, $b_3 g_4$, …, $b_{15} g_{16}$. But the graph also contains the edges $b_5 g_8$ and $b_4 g_6$, and so this matching has augmenting paths of

two different levels starting at the same free girl (Fig. 6):

$$\pi_2 = g_7 \text{ — } b_7 \text{ ═ } g_8 \text{ — } b_8 \quad \text{(level 2)}, \qquad \pi_3 = g_7 \text{ — } b_7 \text{ ═ } g_8 \text{ — } b_5 \text{ ═ } g_6 \text{ — } b_4 \quad \text{(level 3)}. \tag{3.3}$$

So the hypothesis of 15(a), that every augmenting path has the same level, already fails at the first round. On the other hand, Appendix A gives explicit input orders for this graph under which Algorithm H follows the schedule $8, 5, 3, 2, 1, 0$ and passes through exactly the matrices of Fig. 5.

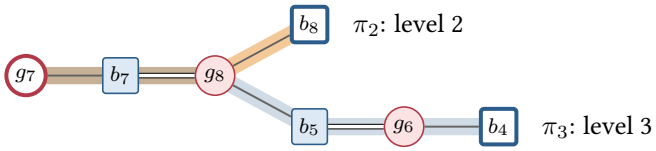


Figure 6. Part of Knuth's 16-pair graph after H1. Both π_2 (level 2, three edges) and π_3 (level 3, five edges) are augmenting paths.

3.4 Three readings of the definition

Exercise 15(c) calls G a Tchoukaillon graph if it might cause Algorithm H "to continually satisfy assumption (a)" while following the schedule. Since the algorithm is deterministic, "might" must refer to the input order of Section 2.3. The example of Fig. 6 shows that the matrices of answer 15(d) provide something weaker than assumption (a). We therefore name three properties that a round with f free girls at level L may have:

- U: every augmenting path has level L (the literal hypothesis of 15(a); U for "uniform");
- F: there are f vertex-disjoint SAPs, that is, $\nu = f$; equivalently, for some perfect matching all f basic paths are SAPs (what the matrices show; F for "full");
- O: no condition beyond the numbers.

Definition 3.2. An ordered graph with a perfect matching *realizes the Tchoukaillon schedule* if H1 leaves $\lfloor s/2 \rfloor$ free girls and every later round has the level L and the number of SAPs a_L prescribed by (3.1). A realization is *literal* if every round after H1 has property U, and *basic-row* if every such round has property F. We call a realization with no extra condition *operational*.

Property U implies F, since the basic paths are augmenting paths. Property F is exactly what Lemma 3.1(b) needs to force $\lceil f/(L+1) \rceil$ SAPs. Neither implication reverses: Knuth's graph is a basic-row realization that is not literal, and Section 8.2 gives an operational realization (an 8-pair tree) in which a round has two free girls but only one SAP, so it is not basic-row.

Which reading did Knuth intend? The literal one is what 15(a) and 15(c) say. The matrices of 15(d) support the basic-row reading, and the motivation—making Algorithm H slow—supports the operational one; the abstract of Knuth's 2022 lecture on Tchoukaillon numbers [12] poses the matching question in terms of running time. These notes do not choose. They answer the literal question completely (Section 7), and report partial results on the other two (Section 8).

Finally there is the question raised in the answer to exercise 16: whether some graph forces *more* rounds than a Tchoukaillon graph would, "by violating assumption (a) and/or by finding more SAPs." Here no condition is imposed on the run at all. Let

$$R(s) = \max \{ \text{number of rounds of Algorithm H} \},$$

the maximum over all ordered bipartite graphs whose maximum matching has size s. (Neither a perfect matching nor any particular initial matching is required.) Sections 4–6 determine $R(s)$ exactly. Table 1 summarizes the four questions and what these notes establish about them.

Question	Requirement	Answer here
Literal	schedule, and U in every round	for $s \geq 1$, exists exactly when $s \in \{1, 2, 3, 4, 5, 8, 9\}$ (Corollary L)
Basic-row	schedule, and F in every round	explicit inputs for all $s \leq 47$ (Appendix C), including Knuth's graph; open in general
Operational	schedule only	the same inputs; open in general; necessary conditions in Section 8
Most rounds	nothing	$R(s) = \lfloor\sqrt{4s+1}\rfloor - 1$ (Theorem R); always $R(s) \geq T(s)$, with strict inequality first at $s = 9$ and for all large s (Section 6)

Table 1. Four questions suggested by exercises 15 and 16.

4 How many rounds can there be?

Here is the main theorem. Recall that rounds are counted with H1 as the first one (Section 2.5), and let

$$Q_n = \left\lfloor \frac{(n+1)^2}{4} \right\rfloor = \left\lfloor \frac{n+1}{2} \right\rfloor \left\lceil \frac{n+1}{2} \right\rceil ,$$

the quarter-squares $(Q_1, Q_2, Q_3, \ldots) = (1, 2, 4, 6, 9, 12, 16, 20, 25, 30, 36, 42, \ldots)$.

Theorem R. *Let $s \geq 0$. Every ordered bipartite graph whose maximum matching has size s makes Algorithm H perform at most*

$$R(s) = \lfloor\sqrt{4s+1}\rfloor - 1$$

rounds, and some ordered graph with $M = N = s$ attains this number. Equivalently, n rounds are possible if and only if $s \geq Q_n$.

The two formulations agree: $Q_n \leq s$ means $(n+1)^2 < 4s+4$, that is $(n+1)^2 \leq 4s+3$; since squares are 0 or 1 modulo 4, this is the same as $(n+1)^2 \leq 4s+1$, or $n \leq \lfloor\sqrt{4s+1}\rfloor - 1$.

This section proves the upper bound. Section 5 constructs graphs that attain it.

4.1 A rectangle that must fit

The argument is the one in the proof of Theorem H, applied at the start of a round instead of at the end.

Lemma 4.1. *Suppose that before some round the current matching has s' couples, and that its shortest augmenting paths have level L. Then*

$$(s - s')\, L \;\leq\; s.$$

Proof. The union of the current matching with a maximum matching contains $s - s'$ vertex-disjoint augmenting paths. Each has level at least L, hence contains at least L edges of the maximum matching. These edges are distinct, and there are only s of them. □

When $s = M = N$, the deficiency $s - s'$ is the number f of free girls, and the lemma says that the f basic paths, drawn as rows of at least L cells each, contain an $f \times L$ rectangle inside the set of all s cells (Fig. 7, left). Nothing requires the rows to have equal numbers of cells; a basic path may have a much higher level than a SAP.

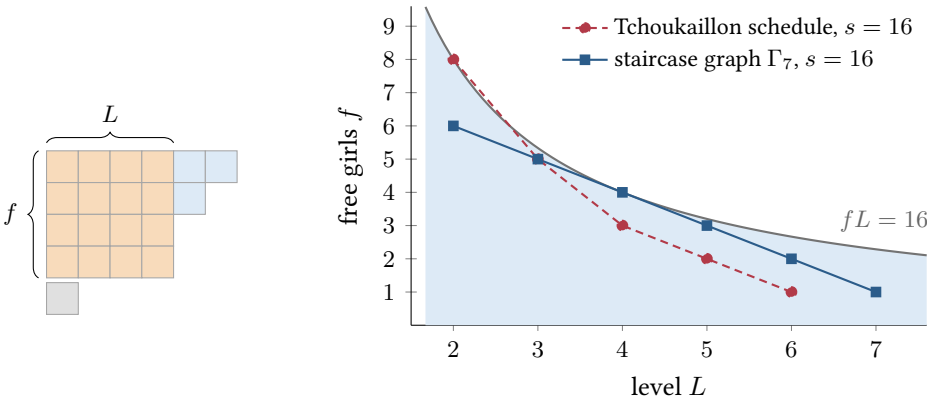


Figure 7. Left: an example with $s = 20$ and $f = 4$. The basic paths of the matching are drawn one per row, each with at least $L = 4$ cells (one cell per pair of the perfect matching); the grey cell lies on no basic path. The shaded $f \times L$ rectangle shows why $fL \le s$. Right: every round of every run is a point (L, f) on or below the hyperbola $fL = s$. Both trajectories shown take $s = 16$; the staircase graph of Section 5 starts with fewer free girls but loses only one per round, and so lasts longer.

4.2 Proof of the upper bound

Suppose a run has $n \ge 2$ rounds, so that there are $n - 1$ successful rounds after H1. Number these $i = 1, 2, \ldots, n-1$ and let round i have level L_i and start with deficiency $d_i = s - s'$.

- $L_i \ge i + 1$. Indeed $L_1 \ge 2$ because the matching produced by H1 is maximal, and the levels strictly increase from round to round.
- $d_i \ge n - i$. Indeed each of the rounds $i, i+1, \ldots, n-1$ adds at least one couple.

By Lemma 4.1, $s \ge (i+1)(n-i)$ for every i. The two factors add up to $n+1$, so the product is largest when they are as nearly equal as possible: taking $i + 1 = \lceil (n+1)/2 \rceil$ gives $s \ge \lfloor (n+1)/2 \rfloor \lceil (n+1)/2 \rceil = Q_n$. A run with a single round needs an edge, so $s \ge 1 = Q_1$. This proves the upper bound in Theorem R. □

Hopcroft and Karp bounded the number of distinct path lengths, and hence the number of phases, by $2\lfloor \sqrt{s} \rfloor + 2$ [8, Theorem 3], by essentially this argument applied at a single well-chosen phase. Using it at every round gives the exact threshold Q_n. What is new in Theorem R is that this threshold is attained.

4.3 What a run with the most rounds must look like

If $s = Q_n$ and a run has n rounds, then the inequalities in the proof are tight at a middle round i, where $(i+1)(n-i) = Q_n$. That round has level exactly $L = i+1$ and deficiency exactly $f = n-i$, and the f disjoint augmenting paths supplied by a maximum matching are all SAPs, which together use every one of its s edges. So $\nu = f$ there, and in the picture the $f \times L$ rectangle fills all s cells, as in Knuth's matrices. From that round on, every round removes exactly one SAP.

The rounds before the middle are where a worst case can differ from a Tchoukaillon graph. There f can exceed $L + 1$, and then, if ν were equal to f, Lemma 3.1(b) would force at least two SAPs in the round. The staircase graphs of Section 5 avoid this by making some basic paths longer than L: those paths are counted in f but are not SAPs, so $\nu \le L + 1$ and a single SAP can meet all the shortest ones. Fig. 7 (right) shows the effect: the Tchoukaillon schedule starts at the top of the hyperbola and must shed free girls quickly, while the staircase graph starts lower and slides down along it, one free girl per round.

5 Staircase graphs

This section constructs, for every $n \geq 2$, an ordered graph Γ_n with $s = Q_n$ on which Algorithm H performs n rounds, and proves exactly what the algorithm does on it. The graph of Fig. 3 is Γ_5. The idea, already visible in Section 2.4, is to let the basic paths have different levels, and to arrange the input so that the depth-first search always finds one SAP that crosses every shortest basic path. That single path blocks all the others.

5.1 The graph and its input

Fix $n \geq 2$. The graph Γ_n is built on a diagram of *cells* arranged in $n-1$ left-justified rows. Row a has

$$\ell_a = \max\Big(2,\ \Big\lfloor \frac{n+2-a}{2} \Big\rfloor\Big) \qquad (1 \leq a \leq n-1)$$

cells, so the numbers of cells decrease by one every two rows, except that the last row has two cells. Equivalently, cell (a, c) (row a, column c) exists if $a + 2c \leq n + 2$, and there is one more cell $(n-1, 2)$, which we call the *corner*. For $n = 5$ the rows have $3, 2, 2, 2$ cells; for $n = 11$ they are $6, 5, 5, 4, 4, 3, 3, 2, 2, 2$ (Fig. 8).

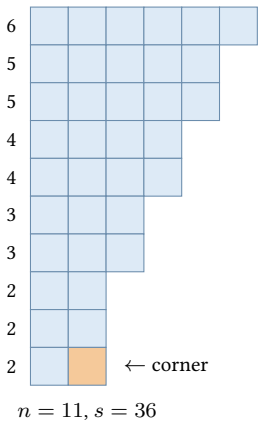


Figure 8. The cell diagram of Γ_{11}; the numbers on the left are the numbers of cells in the rows. The shaded cell is the corner cell $(10, 2)$.

Each cell (a, c) holds a girl $g_{a,c}$ and a boy $b_{a,c}$. The edges are

- $g_{a,c}$ — $b_{a,c}$ for every cell (the *inside* edges);
- $b_{a,c}$ — $g_{a,c+1}$ whenever both cells exist (*horizontal* edges, from a boy to the girl of the next cell in his row);
- $b_{a,c}$ — $g_{a+1,c}$ whenever both cells exist (*vertical* edges, from a boy to the girl of the cell below), *except* the single edge $b_{n-2,2}$ — $g_{n-1,2}$ into the corner, which is omitted.

The omission is a convenience for the proofs. With that edge present, Algorithm H still performs n rounds of one SAP each (we have checked this for $n \leq 30$), but in every round with at least two free girls the corner girl would join the list BLINK of a free boy, which breaks the clean description in Lemma 5.5 and changes the counts of Theorem W. So a girl is adjacent to at most three boys: the boy to her left, the boy above her, and her own. In Fig. 3 girls sit at the upper left of their cells and boys at the lower right, so horizontal edges run up and to the right, and vertical edges down and to the left.

The input order. Number the boys $1, 2, \ldots, s$ in row-major order of their cells (row by row, left to right). Number the first girl of each row, $g_{a,1}$, as $n - a$; number the remaining girls $n, n+1, \ldots, s$ in row-major order. Each girl's list consists of

$$\text{her left boy } b_{a,c-1}, \quad \text{then her upper boy } b_{a-1,c}, \quad \text{then her own boy } b_{a,c},$$

omitting those that do not exist or are not adjacent to her. For $n = 5$ this gives the lists (2.2).

Proposition 5.1. *The graph Γ_n is connected and planar, and has maximum degree three. Its inside edges form a perfect matching, and*

$$M = N = s = Q_n, \qquad E = 3Q_n - n - \left\lfloor \frac{n+1}{2} \right\rfloor .$$

Step H1 *couples every horizontal edge $b_{a,c} == g_{a,c+1}$. It leaves free exactly the first girl and the last boy of every row, and puts the free girls into* QUEUE *in row order $g_{1,1}, g_{2,1}, \ldots, g_{n-1,1}$. It examines $s + n - 2$ adjacency entries.*

Proof. Substituting $j = n + 2 - a$, the numbers of cells in the rows are $\max(2, \lfloor j/2 \rfloor)$ for $3 \le j \le n + 1$. Since $\sum_{j=1}^{m} \lfloor j/2 \rfloor = \lfloor m^2/4 \rfloor$, their sum is $\lfloor (n+1)^2/4 \rfloor - \lfloor 1/2 \rfloor - \lfloor 2/2 \rfloor + 1 = Q_n$. There are s inside edges, $s - (n-1)$ horizontal edges, and one vertical edge into every cell outside row 1 except the corner, so $s - \ell_1 - 1$ vertical edges when $n \ge 3$; with $\ell_1 = \lfloor (n+1)/2 \rfloor$ this gives E (and $n = 2$ is checked directly). Girls have at most three neighbors by construction, and so do boys (their own girl, the girl to the right, and the girl below). Rows are connected by their horizontal and inside edges, and the vertical edges in column 1 connect the rows. The drawing of Fig. 3 has no crossings, because every edge stays inside its own cell or in the gap between two adjacent cells.

Step H1 scans girls $s, s-1, \ldots, n$ first; these are the girls not in column 1, and each takes the first boy on her list, her left boy. All these boys are different. Then girls $n-1, \ldots, 1$, the first girls of rows $1, \ldots, n-1$, find their upper and own boys taken. The count of examinations is $(s - n + 1) + 1 + 2(n - 2)$. □

5.2 Watching the smallest interesting case

Before the general proof, look at the whole run on Γ_5 (Fig. 9). The coloured bands are the basic paths relative to the inside-edge perfect matching. Initially they are the rows of the diagram, with $3, 2, 2, 2$ cells, that is, of levels $3, 2, 2, 2$. The three basic paths of level 2 are disjoint SAPs, so the dag has $L = 2$ and $\nu = 3$, while $f = 4$. Yet the single SAP that Algorithm H finds runs from the free girl of row 4 up through row 3 to the free boy of row 2, meeting all three. After augmenting there are three basic paths, each with three cells: the full rectangle that Section 4.3 demands. The next SAP climbs a staircase through all three of them, and so on.

L	f	SAPs	W_5	W_{13}	W_{11}	the SAP, as couples (g, b) in path order
2	4	1	17	5	8	$(1,6), (8,5)$
3	3	1	19	8	9	$(2,4), (7,2), (6,3)$
4	2	1	17	8	8	$(3,1), (5,2), (7,5), (8,7)$
5	1	1	9	5	5	$(4,1), (3,4), (2,6), (1,8), (9,9)$

The SAPs are listed by their newly formed couples, using the numerical labels of Fig. 3; for example $(1,6), (8,5)$ is g_1 — b_6 == g_8 — b_5. Step H1 examined 12 adjacency entries, so the run costs $12 + 62 + 26 + 30$ elementary operations in all.

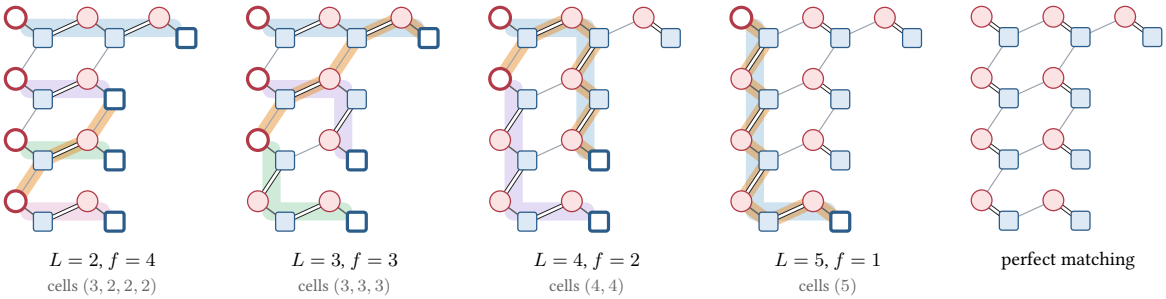


Figure 9. The run of Algorithm H on Γ_5. Each panel shows the matching before one round, drawn as in Fig. 3 but without labels; coloured bands are the current basic paths, and the orange band is the one SAP that the round finds. The graph never changes, only the matching. The basic paths get longer and fewer; their numbers of cells go $(3, 2, 2, 2) \to (3, 3, 3) \to (4, 4) \to (5)$.

5.3 Hooks

To describe the basic paths in general, call the state with f free girls *state* f; in it the dag will have level

$$L = n + 1 - f.$$

The run goes through states $f = n - 1, n - 2, \ldots, 1, 0$.

Definition 5.2. In state f, for $1 \le x \le f$, the *hook* x is the following sequence of cells. Let $c = f + 1 - x$ be its *corner column*.

- If row x has fewer than c cells, hook x is row x.
- Otherwise hook x consists of $(x, 1), (x, 2), \ldots, (x, c)$, followed by the cells below (x, c) in column c, from top to bottom, except the corner cell.
- The last hook, $x = f$, ends with the corner cell $(n - 1, 2)$.

So a hook runs east along its row to its corner column and then turns south (Fig. 10). Let M_f be the matching that couples the boy of each cell of a hook to the girl of the next cell of the same hook, and couples the girl and boy of every cell that is in no hook. A cell in no hook is called *retired*; the others are *alive*. In M_f the free girls are the first girls of the hooks, the free boys are the last boys of the hooks, and the basic paths with respect to the inside edges are precisely the hooks. Let M_0 be the perfect matching of inside edges.

Lemma 5.3. *Let* $1 \le f \le n - 1$ *and* $L = n + 1 - f$.

(a) The hooks are disjoint, and their cells are exactly the cells in columns $1, \ldots, f$ *together with the corner cell. An alive cell* (a, c) *other than the corner lies in hook* $x = \min(a, f + 1 - c)$*, at position* $y = c + \max(0, a + c - f - 1)$ *along it; the corner is at position* L *of hook* f.

(b) Consecutive cells of a hook are joined by a horizontal or vertical edge of Γ_n*, so* M_f *is a matching of* Γ_n.

(c) Hook x *has* $\max(\ell_x, L)$ *cells. Hence hooks* $1, \ldots, k-1$ *have more than* L *cells and hooks* $k, \ldots, f$ *have exactly* L *cells, where*

$$k = \max(1, f - L), \qquad m = f - k + 1 = \min(f, L + 1).$$

(d) M_{n-1} *is the matching produced by* H1.

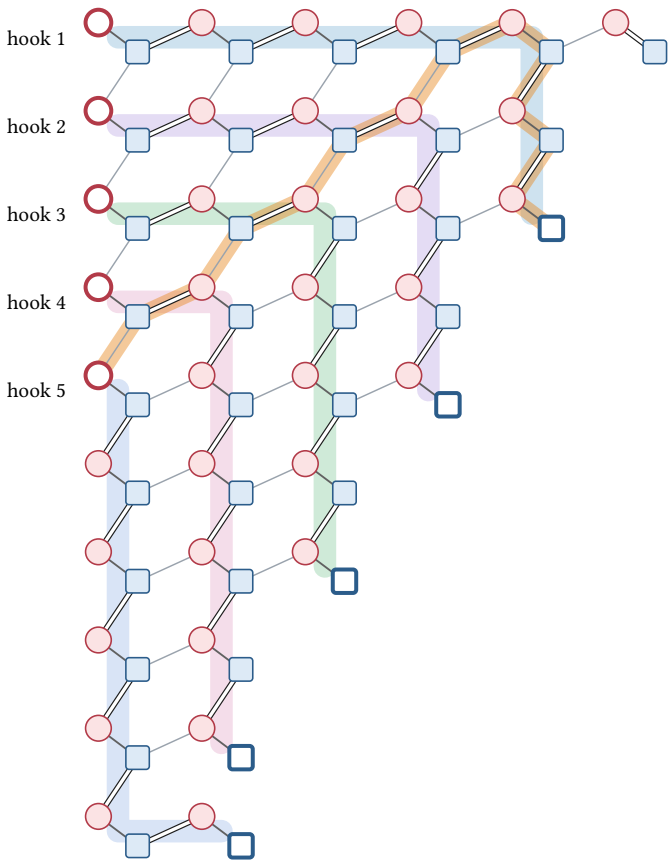


Figure 10. State $f = 5$ of Γ_{11}, before the round at level $L = 7$. The five hooks (coloured) run east along their rows and turn south at columns $5, 4, 3, 2, 1$. Cells in columns 6 and beyond are retired. All five hooks have 7 cells. For instance cell $(7, 2)$ lies in hook $\min(7, 6 - 2) = 4$, at position $2 + \max(0, 7 + 2 - 6) = 5$, as Lemma 5.3(a) says. The orange staircase is the SAP that Algorithm H finds: it starts at the free girl of hook 5 and climbs to the free boy of hook 1.

Proof. (a) A cell (a, c) with $c \le f$ and $a + c \le f + 1$ lies in row a at or before the corner column $f + 1 - a$ of hook a, so it belongs to hook a at position c. If $a + c > f + 1$, the hook whose corner column is c is hook $x = f + 1 - c < a$; since columns are top-justified, cell (x, c) exists, so hook x turns south at (x, c) and contains (a, c) at position $c + (a - x)$, unless (a, c) is the corner. Conversely every hook stays in columns $\le f$ apart from the corner, and the two cases above assign each cell to only one hook.

(b) East steps use horizontal edges and south steps use vertical edges; the omitted edge would lead into the corner, which is entered from its left neighbour $(n - 1, 1)$ instead.

(c) Write $c = f + 1 - x$, and note that $2\ell_x \ge n + 1 - x$ always. If $\ell_x < c$, hook x is row x; then $x \le f - \ell_x$, so $2\ell_x \ge n + 1 - f + \ell_x$, that is, $\ell_x \ge L$. Otherwise, if $x < f$, hook x consists of c cells of row x followed by the cells of column c below it. Column $c \ge 2$ ends at row $n + 2 - 2c$ (the corner is not counted), so the hook has $c + (n + 2 - 2c - x) = n + 2 - c - x = L$ cells. The last hook has $1 + (n - 1 - f) + 1 = L$ cells. In this second case $\ell_x \le L$ as well: for $x = n - 1$ we have $\ell_x = 2 = L$, and otherwise $2\ell_x \le n + 2 - x$, whereas $\ell_x \ge L + 1$ and $\ell_x \ge c$ would give $2\ell_x \ge n + 3 - x$. Finally, $\ell_x \ge L + 1 = n + 2 - f$ if and only if $n + 2 - x \ge 2(n + 2 - f)$, that is, $x \le f - L - 1$.

(d) For $f = n - 1$ the corner column of hook x is $n - x$, which exceeds ℓ_x when $x \le n - 3$; for the last two rows the definition also gives the whole row. So M_{n-1} couples the horizontal edges, as in Proposition 5.1. □

Write $y(a, c)$ for the position of an alive cell, as given by part (a). Two facts about it drive the next two lemmas: y never decreases when we move from an alive cell to the alive cell below it or to its right, and it increases by exactly one from each cell of a hook to the next. (For the corner, $y = L$; its left neighbour has position $L - 1$, and the cell above it, $(n - 2, 2)$, also has position L.)

5.4 No shortcuts

The graph has many edges that the hooks do not use, and some of them will be needed in later rounds. The next lemma shows that none of them creates a shorter augmenting path now.

Lemma 5.4. *In the matching M_f, the girl of an alive cell at position y of its hook is at distance $2(y-1)$ from the free girls, and its boy at distance $2y-1$. No retired vertex can be reached by an alternating path from a free girl. Consequently the dag has level L, and its SAPs end at the free boys of hooks $k, \ldots, f$.*

Proof. First, retired vertices. A retired boy $b_{a,c}$ can be entered only from the girl to his right or the girl below him, since his own girl is his mate. Both lie in retired cells, being in column c or $c + 1$, beyond column f. (When $f = 1$ the corner is alive and lies below the retired cell $(n - 2, 2)$; this is one place where the omitted edge matters.) A retired girl can be entered only from her own boy. So no alternating path from a free girl ever reaches a retired vertex.

Now follow an alternating path from a free girl. A step from a girl to a boy uses an edge outside M_f, which goes from cell (a, c) to the same cell, to $(a, c - 1)$, or to $(a - 1, c)$; by the monotonicity of positions it does not increase the position. A step from a boy to a girl uses a couple of M_f, and so increases the position by exactly one. The path starts at position 1, so a boy at position y is reached only after at least $y - 1$ couples, that is, after at least $2y - 1$ edges. The beginning of his own hook reaches him in exactly that many. The argument for girls is the same.

Free boys are the last boys of the hooks, at positions $\max(\ell_x, L) \geq L$. The smallest distance to one of them is therefore $2L - 1$, attained exactly by hooks $k, \ldots, f$. □

5.5 The forced search

Now the input order enters. Fig. 11 shows a dag of a staircase graph with each hook drawn as a straight row; the positions become the columns, which are exactly the levels of the dag.

Lemma 5.5. *Suppose a round starts with the matching M_f and with the free girls in* QUEUE *in hook order. Then each level G_j of the dag is listed in hook order, and for the boy b at position $y \leq L$ of hook x,*

$$\texttt{BLINK}[b] = \begin{cases} (\textit{girl at position } y \textit{ of hook } x+1,\ \textit{girl at position } y \textit{ of hook } x), & \textit{if } x + y \leq f; \\ (\textit{girl at position } y \textit{ of hook } x), & \textit{if } x + y > f. \end{cases}$$

The roots of the depth-first search are the free boys of hooks $k, k + 1, \ldots, f$, in this order.

Proof. Let $b = b_{a,c}$ lie at position y of hook x. By Lemma 5.4, the girls one level below b are those at position y. Of his three possible neighbours, his own girl qualifies; the girl to his right does not, since her position is larger; and the girl below him, $g_{a+1,c}$, if she is adjacent to him, qualifies exactly when the two cells have the same position. By the formula of Lemma 5.3(a) that means $a + c \leq f$, with one exception: the corner has the same position as the cell above it, but the edge between them is the omitted one, which is the main reason for omitting it. When $a + c \leq f$, then (a, c) lies in the first part of hook a, so $c = y \leq L$, and $(a + 1) + 2c \leq f + 1 + L = n + 2$; hence cell $(a + 1, c)$ exists, and it is not the corner because $a + c \leq f < n$. Also (a, c) is at position c of hook a, so $a + c \leq f$ is the condition $x + y \leq f$, and $g_{a+1,c}$ is at position y of hook $x + 1$.

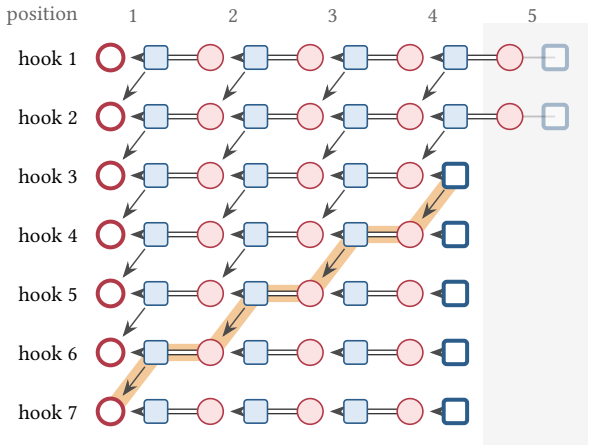


Figure 11. The dag of the round at level $L = 4$ of Γ_{10} (state $f = 7$), with the hooks straightened. Arrows are the arcs of BLINK, from a boy to a girl one level down: every boy has an arc to his own girl, and the boy at position y of hook x also has an arc to the girl below him when $x + y \le 7$; that arc comes first in BLINK. Hooks 1 and 2 have more than L cells (the shaded column lies beyond the dag). The search starts at the free boy of hook 3, the first hook with exactly L cells, and always takes the first arc, so it goes down a staircase to the last free girl.

Now argue by induction on j that G_j is in hook order. For $j = 0$ this is the hypothesis. When the girls of G_j are scanned, the girl of hook x meets at most three boys: her own boy; her left boy, whose position is smaller, so that he is already in the dag at a lower level and step H6 ignores him; and her upper boy, whose position is at most hers. (On the vertical part of a hook her upper boy is her mate, at the level below.) The only girls of G_j adjacent to her own boy are herself and the girl of hook $x + 1$, who comes later; so she is the one who discovers him. Thus B_{j+1} is discovered in hook order, and so is G_{j+1}, which consists of their mates. When the girl of hook $x + 1$ later meets the same boy as her upper boy, step H6 inserts her at the head of his list, which gives the stated order of BLINK.

At level L the boys of hooks $1, \ldots, k-1$ are matched and are discovered before the first free boy (in hook k), so they enter the dag; the free boys of hooks $k, \ldots, f$ follow in order and become the roots. □

Lemma 5.6. *Under the hypotheses of Lemma 5.5, the round finds exactly one SAP. Its free girl is the last girl in* QUEUE*, and afterwards the matching is M_{f-1}, with the remaining free girls still in hook order.*

Proof. Write $\langle x, y\rangle$ for position y of hook x, and recall that hooks $k, \ldots, f$ have exactly L cells and $f - k \le L$.

The first root. The search starts at the boy $\langle k, L\rangle$ and always follows the first remaining arc. Suppose first that $f \ge L + 1$, so that $k = f - L$ and $k + L = f$. Then the first arc leads to the girl $\langle k + 1, L\rangle$; her mate is the boy $\langle k + 1, L - 1\rangle$, whose hook and position again add up to f, so the next arc leads to the girl $\langle k + 2, L - 1\rangle$, and so on down the staircase

$$\text{boy } \langle i, f-i\rangle \;\to\; \text{girl } \langle i+1, f-i\rangle \;=\!=\; \text{boy } \langle i+1, f-i-1\rangle \qquad (i = k, k+1, \ldots, f-1),$$

ending at the girl $\langle f, 1\rangle$, who is free. If instead $f \le L$, then $k = 1$, and the search first walks backwards along hook 1 (each boy $\langle 1, y\rangle$ with $1 + y > f$ has only his own girl) until it reaches the boy $\langle 1, f - 1\rangle$; from there it descends the same staircase. In both cases the SAP ends at the free girl of hook f, the last girl in QUEUE, so step H16 leaves the order of the others unchanged.

The other roots fail. The SAP passes through the girl $\langle i, f + 1 - i\rangle$ of every hook i with $k < i \le f$. The root $\langle i, L\rangle$ of such a hook has only one arc at each position y with $i + y > f$, namely to its own

girl. Following these single arcs, it reaches the boy $\langle i, f+1-i\rangle$, whose only girl has been deleted. Every list on the way is now empty, and step H15 backs up all the way: the root fails. So the round has exactly one SAP.

The new matching. Augmenting dissolves, in each hook i with $k < i < f$, the couple between positions $f-i$ and $f+1-i$, and couples instead the boy at position $f-i$ of hook i to the girl at position $f-i$ of hook $i+1$, for $k \le i < f$. Reading along the new couples, the new hook i therefore consists of the first $f-i$ cells of the old hook i, followed by the cells of the old hook $i+1$ from position $f-i$ on; the remaining cells of the old hook i have passed to hook $i-1$, and the old hook f has been absorbed into hook $f-1$. When $f \le L$, the cells of hook 1 at positions $f, \ldots, L$, which the search walked through, become coupled to themselves, that is, retired. Now compare with the definition. In state f, hook $i+1$ has corner column $f-i$, so the new hook i runs east along row i to column $f-i$ and then south: it is hook i of state $f-1$. The cells that retire are the alive cells of column f other than the corner (which stays alive when $f=2$). Hooks $1, \ldots, k-1$ are untouched whole rows, and they are whole rows in state $f-1$ too: for them $\ell_x \ge L+1 = n+2-f$, so their new corner column satisfies $f-x \ge n+2-x-\ell_x \ge \ell_x$, and if equality holds then $x+2\ell_x = n+2$ and there is no cell below (x, ℓ_x). So the new matching is M_{f-1}. For $f=1$ the search walks back along the single hook and produces the perfect matching M_0. □

Searched on its own, the root of each of the hooks $k+1, \ldots, f$ would find a SAP along its own hook. But `MARKED` lists the roots in hook order, so the root of hook k goes first, and because the first arc of each `BLINK` list points one hook down, its path takes the girl at position $f+1-i$ from every later hook i. Each later root must pass through that girl, so it fails.

Theorem S. *On the ordered graph Γ_n, Algorithm H performs exactly n rounds. For $L = 2, 3, \ldots, n$, the round at level L starts with $f = n+1-L$ free girls and finds exactly one SAP. The final matching consists of the inside edges.*

Proof. By Proposition 5.1 and Lemma 5.3(d), step H1 produces M_{n-1} with the free girls in hook order. Apply Lemmas 5.4–5.6 for $f = n-1, n-2, \ldots, 1$. □

Every size. Theorem S gives n rounds when $s = Q_n$. For $Q_n < s < Q_{n+1}$, append $p = s - Q_n$ new girl–boy pairs to Γ_n, numbered after all the old ones. Give each new girl the list "her own new boy, then the previous boy," where the previous boy is the corner boy $b_{n-1,2}$ for the first new girl and the preceding new boy afterwards. Step H1 scans the new girls first and couples each to her own boy. No old girl's list mentions a new boy, so an alternating path from an old free girl never enters the new pairs: to do so it would have to arrive at a boy whose mate is a new girl. Hence every round runs exactly as on Γ_n. The corner boy had degree one, so the new graph is still planar, connected, and of maximum degree three. For $s=1$ use a single edge, and for $s=0$ the empty graph. Together with Section 4, this proves Theorem R. □

5.6 How much one path blocks

In state f the hooks $k, \ldots, f$ are $m = \min(f, L+1)$ disjoint SAPs, so the largest set of disjoint SAPs has $\nu = m$ members, while Algorithm H takes one. The total number of SAPs in the dag grows exponentially in m.

Proposition 5.7. *In state f, the dag of Γ_n contains exactly $2^m - 1$ SAPs.*

Proof. Let $N(x, y)$ be the number of dag paths from the boy at position y of hook x down to $\perp$. By Lemma 5.5, $N(x,y) = N(x, y-1) + [x+y \le f]\, N(x+1, y-1)$, where $N(x,0) = 1$ stands for the free girl. Induction on y gives $N(x,y) = 2^{\min(y, f-x)}$. A root $\langle i, L\rangle$ with $k \le i \le f$ therefore starts 2^{f-i} SAPs, since $f-i \le f-k \le L$, and $\sum_{i=k}^{f} 2^{f-i} = 2^m - 1$. □

Every SAP from the root of hook $i > k$ passes through the girl at position $f + 1 - i$ of that hook, which the first SAP uses. So the one SAP found by Algorithm H meets all $2^m - 1$ SAPs, including all m disjoint ones. Lemma 3.1(b) with $\nu = m \le L + 1$ in place of f predicts at least one SAP, and one is what Algorithm H finds.

5.7 Counting the work

Theorem W. *In the round of Γ_n with f free girls and level $L = n + 1 - f$, with $m = \min(f, L + 1)$ as above,*

$$W_5 = 3fL - n - \min(f, L), \qquad W_{13} = mL - \binom{m-1}{2}, \qquad W_{11} = fL.$$

Summed over all rounds after H1*,*

$$\sum W_5 = \frac{(n-1)n(n+2)}{2} - Q_n + 1, \qquad \sum W_{11} = \frac{(n-1)n(n+4)}{6},$$

$$\sum W_{13} = \begin{cases} (4u^3 + 21u^2 + 23u - 6)/6, & n = 2u + 1, \\ (4u^3 + 15u^2 + 5u - 12)/6, & n = 2u. \end{cases}$$

The final, unsuccessful search examines no edges.

Proof. The breadth-first search scans the girls at positions $1, \ldots, L$ of all f hooks, fL girls in all, and every list entry of each. A girl has three neighbours except that girls in column 1 have no left boy, girls in row 1 have no upper boy, and the corner girl lacks the omitted edge. The scanned girls include all $n - 1$ girls of column 1, exactly $\min(f, L)$ girls of row 1, and the corner girl; this gives W_5. The dag contains all fL boys at positions $\le L$, by the last paragraph of the proof of Lemma 5.5, so $W_{11} = fL$. The first root consumes L arcs. The root of hook $k + h$, $1 \le h < m$, consumes the arcs at positions $L, L-1, \ldots, m-h$, including the one to a deleted girl: $L - m + h + 1$ arcs. Adding up gives W_{13}, and summing over $f = 1, \ldots, n - 1$ gives the totals (the sum $\sum_f \min(f, n + 1 - f)$ equals $Q_n - 1$). □

For Γ_5 the totals are $\sum W_5 = 62$, $\sum W_{13} = 26$, $\sum W_{11} = 30$, in agreement with the table of Section 5.2. As $n \to \infty$ we have $\sum W_5 \sim n^3/2$, $E \sim 3n^2/4$, and $V = 2Q_n \sim n^2/2$, so

$$\frac{\sum W_5}{E\sqrt{V}} \longrightarrow \frac{2\sqrt{2}}{3} \approx 0.943.$$

Thus the breadth-first searches alone examine $\Theta(E\sqrt{V})$ adjacency entries, and Corollary K is tight up to a constant factor, even for connected planar graphs of maximum degree three. The padding used for other sizes adds only $O(n)$ edges and does not change this.

6 The true maximum against the Tchoukaillon schedule

We now have two staircases of thresholds (Fig. 1): n rounds are possible for some graph exactly when $s \ge Q_n$, and a Tchoukaillon graph, if one exists, has n rounds exactly when $Ч_n \le s < Ч_{n+1}$:

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Q_n	1	2	4	6	**9**	12	**16**	**20**	**25**	**30**	**36**	**42**	**49**	**56**
$Ч_n$	1	2	4	6	10	12	18	22	30	34	42	48	58	60

(Bold entries are those with $Q_n < Ч_n$.)

6.1 Always $T(s) \le R(s)$

Lemma 6.1. *$Q_n \le \mathrm{Ч}_n$ for all $n \ge 1$. Consequently $T(s) \le R(s)$ for all s.*

Proof. Run the schedule (3.1) with $s = \mathrm{Ч}_n$, so that it has n rounds and $f_L > 0$ for $2 \le L \le n$. By induction $f_L \le s/L$: indeed $f_2 = \lfloor s/2 \rfloor$, and $f_{L+1} = \lfloor L f_L/(L+1) \rfloor \le s/(L+1)$. On the other hand each of the levels $L, L+1, \ldots, n$ removes at least one free girl, so $f_L \ge n+1-L$. Hence $s \ge L(n+1-L)$ for $2 \le L \le n$, and choosing $L = \lceil (n+1)/2 \rceil$ gives $s \ge Q_n$ when $n \ge 3$; the cases $n \le 2$ are in the table. Since $T(s) = n$ means $s \ge \mathrm{Ч}_n \ge Q_n$, Theorem R gives $R(s) \ge n$. □

This is the argument of Section 4 applied to the numbers alone; it does not assume that any Tchoukaillon graph exists.

6.2 Where the true maximum is larger

By Lemma 6.1, $R(s) > T(s)$ exactly when $Q_n \le s < \mathrm{Ч}_n$ for $n = R(s)$, and $R(s) = T(s)$ exactly when $\mathrm{Ч}_n \le s < Q_{n+1}$ for some n. The second interval is nonempty only when $\mathrm{Ч}_n < Q_{n+1}$. Computing $\mathrm{Ч}_n$ for all $n \le 10^5$ by the formula of exercise 11(c), we find that this happens only for $n \le 8$ and $n = 10, 12, 14$. Hence:

- $R(s) = T(s)$ for $s \le 8$ and for $s \in \{10, \ldots, 15,\ 18, 19,\ 22, 23, 24,\ 34, 35,\ 48,\ 60, \ldots, 63\}$;
- $R(s) > T(s)$ for every other s with $9 \le s < \mathrm{Ч}_{100001} \approx 3.2 \times 10^9$, and for all sufficiently large s.

The last statement follows from the asymptotic formulas: $R(s) = 2\sqrt{s} + O(1)$, while exercise 11(d) gives $T(s) = \sqrt{\pi s} + O(s^{1/6})$, so

$$R(s) - T(s) = (2 - \sqrt{\pi})\sqrt{s} + O(s^{1/6}), \qquad \frac{R(s)}{T(s)} \to \frac{2}{\sqrt{\pi}} \approx 1.128.$$

(We have not tried to make "sufficiently large" explicit; doing so would need an explicit constant in the error term of exercise 11(d).) Fig. 12 shows the two normalized counts.

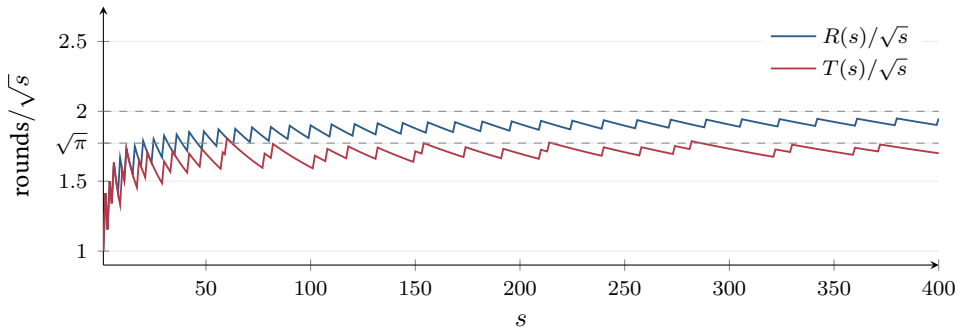


Figure 12. $R(s)/\sqrt{s}$ and $T(s)/\sqrt{s}$ for $s \le 400$. They tend to 2 and $\sqrt{\pi}$. Both are computed from the exact formulas: Theorem R and the schedule (3.1).

6.3 Why nine is the first size

Five rounds need dag rounds at levels $2, 3, 4, 5$ that start with at least $4, 3, 2, 1$ free girls, and the round at level 3 then needs $3 \times 3 = 9$ cells (Lemma 4.1). At $s = 8$ one sees concretely where the attempt breaks down. Five rounds at $s = 8$ would need four free girls after H1, the most possible since $\lfloor 8/2 \rfloor = 4$; so suppose H1 leaves four girls free, and fix a perfect matching. Each of the four

basic paths contains at least two edges of the perfect matching, because the matching from H1 is maximal, and there are only eight such edges; so all four basic paths have level 2, and $\nu = f = 4$. By Lemma 3.1(b) the first round removes at least two SAPs, and only two free girls remain for the rounds at levels $3, 4, \ldots$.

With one more pair, one basic path can have level 3 while the other three have level 2. Now $\nu = 3$, and a single SAP can meet all three basic paths of level 2. This is exactly the first state of Γ_5, whose basic paths have $3, 2, 2, 2$ cells. In every later state of Γ_n with $f > L + 1$ the same thing happens: the hooks $1, \ldots, k-1$ have more than L cells, so $\nu = L + 1 < f$.

At $s = 16$, the size of Knuth's example, Theorem R gives seven rounds, one more than a Tchoukaillon graph; Γ_7 attains them.

6.4 Initial deficiency

A Tchoukaillon graph starts with the largest possible number of free girls, $\lfloor s/2 \rfloor$. The staircase graph Γ_n starts with only $n - 1 \approx 2\sqrt{s}$ of them: step H1 has already found almost all of the matching. Yet it takes more rounds, because it spends them on augmenting paths of many different levels, one at a time. This also means that Theorem R says nothing about the operational question, which insists on the initial deficiency $\lfloor s/2 \rfloor$. An intermediate question is the maximum number of rounds when H1 is required to leave $\lfloor s/2 \rfloor$ girls free; we have not determined it.

7 The literal definition

This section settles the literal reading of exercise 15(c), in which every augmenting path has the current shortest level at every round. Two consecutive rounds of this kind lose at least half of the free girls, while the Tchoukaillon schedule asks the first round at level 2 to remove only about a third of them.

7.1 Uniform consecutive levels force halving

Throughout this subsection the graph has a perfect matching ($s = M = N$), and distances are measured in the oriented graph of Section 2.2.

Theorem U. *Suppose a matching M' with f free girls has f vertex-disjoint SAPs, of level L (property* F*; it holds in particular under property* U*). Let M'' be obtained from M' by augmenting along $a \geq 1$ vertex-disjoint SAPs. If every augmenting path for M'' has level $L + 1$, then*

$$f - a \leq a.$$

In particular, if a round has property F *and the next round has property* U *at the next level, the first of the two rounds removes at least half of the free girls.*

Proof. Call the a paths along which we augmented the *selected* paths. On a selected path $S = g_0 - b_1 = g_1 - \cdots = g_{L-1} - b_L$, write e_i for its edge $g_{i-1} - b_i$ outside M'; it is a couple of M''.

Step 1: old distances are still lower bounds. For M', let $d(v)$ be the distance of v from the free girls, and $h(v)$ the length of a shortest directed path from v to a free boy. On a selected path these are exact: $d(g_i) = 2i$, $d(b_i) = 2i - 1$, $h(g_i) = 2(L - i) - 1$, and $h(b_i) = 2(L - i)$. Passing to M'' reverses the arcs of the selected paths and makes some free vertices matched. The reversed arcs are $b_i \to g_{i-1}$ and $g_i \to b_i$, and they satisfy $d(\text{head}) \leq d(\text{tail}) + 1$ and $h(\text{tail}) \leq h(\text{head}) + 1$: for instance $d(g_{i-1}) = 2i - 2 \leq d(b_i) + 1$ and $h(b_i) = 2(L - i) \leq h(g_{i-1}) + 1$, and similarly for

$g_i \to b_i$. All other arcs are unchanged. So d and h remain lower bounds for the corresponding quantities with respect to M''.

Step 2: contacts are aligned. Let A be an augmenting path for M'' that uses the couple e_i, say as its kth couple counting from its free girl. Its part up to b_i has $2k-1$ edges and its part from g_{i-1} on has $2L+1-2k$ edges, so Step 1 gives $2k-1 \geq 2i-1$ and $2L+1-2k \geq 2(L-i)+1$. Hence $k=i$: a new augmenting path can use e_i only as its ith couple.

Step 3: new basic paths meet the selected paths. Let P^* be the perfect matching obtained from M' by augmenting along all f of its disjoint SAPs. The symmetric difference of M'' and P^* contains $f-a$ vertex-disjoint augmenting paths for M''; call their set $\mathcal{A}$. A member of $\mathcal{A}$ that avoided all selected paths would be a component of $M' \triangle P^*$ as well, that is, one of the old SAPs, still augmenting and of level L, contrary to the hypothesis. So every member of $\mathcal{A}$ meets a selected path, and since all vertices of the selected paths are matched in M'', it contains one of their couples e_i.

Step 4: no selected path meets two members of $\mathcal{A}$. Suppose the selected path S meets two. Among the couples of S that belong to members of $\mathcal{A}$, pick two, $e_i \in A_i$ and $e_j \in A_j$ with $i<j$ and $A_i \neq A_j$, such that no couple strictly between them belongs to a member of $\mathcal{A}$. Now splice (Fig. 13): follow A_j from its free girl through e_j to g_{j-1}, then walk backwards along S through $b_{j-1}, g_{j-2}, \ldots, g_i$ to b_i, then follow A_i from b_i through e_i to its free boy. The three pieces meet only at their ends, so this is an augmenting path for M''. By Step 2 its number of edges is

$$2j \;+\; \big(2(j-i)-1\big) \;+\; \big(2L+2-2i\big) \;=\; 2L+1+4(j-i) \;>\; 2L+1,$$

so its level exceeds $L+1$, contradicting the hypothesis.

By Steps 3 and 4, each of the $f-a$ members of $\mathcal{A}$ meets a selected path, and each of the a selected paths meets at most one of them. So $f-a \leq a$. □

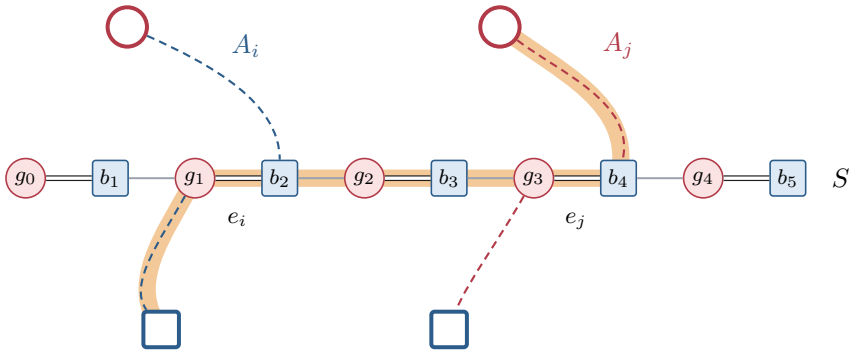


Figure 13. Step 4 of the proof of Theorem U. After augmenting along the selected path S, its couples are the edges e_k from g_{k-1} to b_k. If two disjoint augmenting paths A_i and A_j cross S at e_i and e_j, the shaded splice is an augmenting path with $4(j-i)$ more edges than either of them.

The splice argument also shows that a full set of disjoint new SAPs can coexist with much longer augmenting paths.

Proposition 7.1. *In the situation of Theorem U, drop the hypothesis that all augmenting paths for M'' have level $L+1$, and assume only that its SAPs have level $L+1$. If M'' has q vertex-disjoint SAPs, each meeting a selected path, then M'' has an augmenting path with at least $2L+1+4(\lceil q/a \rceil - 1)$ edges.*

Proof. Steps 1 and 2 apply to SAPs. Some selected path S meets at least $k = \lceil q/a \rceil$ of the q SAPs. Choose a shortest interval $[i,j]$ of indices such that the couples $e_i, \ldots, e_j$ of S include contacts with all k of them. The SAPs owning e_i and e_j have no other contact inside the interval (otherwise it could be shortened), and $j-i \geq k-1$. Splice as in Step 4; contacts with other SAPs inside the interval do no harm, because those paths are not used. □

For example, in the first dag round of Knuth's 16-pair graph three SAPs are selected, and the next matrix has five disjoint SAPs of level 3, each meeting a selected one. So the second matching has an augmenting path with at least $5 + 4 = 9$ edges, that is, of level at least 5; the census in Appendix A shows six of them.

7.2 Consequences

Corollary 7.2. *If property* U *holds in consecutive rounds at levels* $L, L+1, \ldots$*, and the first of them starts with* $f > 0$ *free girls, there are at most* $\lfloor \lg f \rfloor + 1$ *such rounds. The bound is attained when* $f = 1, 2, 4, 8, 16, 32$*, and we expect it to be attained for every power of 2.*

Proof. By Theorem U every round except the last leaves at most $\lfloor f/2 \rfloor$ free girls. Appendix B describes ordered graphs, one for each $f = 2^t$, on which Algorithm H halves the number of free girls in every round while every augmenting path has the current shortest level (Fig. 14). For $f \le 32$ we verified this by running Algorithm H and enumerating all augmenting paths in every round; the appendix explains why it should hold in general, but we have not written out a complete proof. □

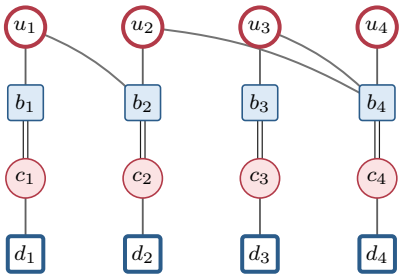


Figure 14. The halving graph for $f = 4$ ($s = 8$), with the matching after H1. Free girl u_i is adjacent to b_i and, if $i < 4$, also to $b_{i+2^{\rho(i)}}$, where $2^{\rho(i)}$ is the largest power of 2 dividing i. Each dag round merges pairs of neighbouring groups of indices (Appendix B), and the free girls go $4 \to 2 \to 1 \to 0$. At every stage all augmenting paths have the same level.

Corollary L. *For* $s \ge 1$*, a literal realization of the Tchoukaillon schedule exists if and only if* $s \in \{1, 2, 3, 4, 5, 8, 9\}$.

Proof. The schedule starts with $f = \lfloor s/2 \rfloor$ free girls at level 2 and removes $a = \lceil f/3 \rceil$ SAPs. If any girls remain, the next round is at level 3 and, in a literal realization, has property U; then Theorem U requires $f \le 2\lceil f/3 \rceil$. This fails for $f = 3$ and for every $f \ge 5$, which excludes $s = 6, 7$ and all $s \ge 10$.

For $s = 2, 4, 8$, take the halving graphs with $f = 1, 2, 4$: their free girls go $1 \to 0$, $2 \to 1 \to 0$, and $4 \to 2 \to 1 \to 0$, which is the schedule in these cases. For $s = 3, 5, 9$, add one isolated girl–boy edge, numbered last; H1 matches it and nothing else changes, and $\lfloor s/2 \rfloor$ is unchanged. For $s = 1$ a single edge will do. □

So Knuth's 16-pair graph cannot be repaired into a literal example: no graph and no input order satisfy the literal definition at $s = 16$. Nothing in this section obstructs the basic-row or operational readings, and Knuth's graph meets the basic-row reading at $s = 16$.

8 The operational question

Can every s be given an ordered graph that realizes the Tchoukaillon schedule, with no condition on the paths? This is exercise 16 under the operational reading, and it is open. This section collects

what we know: explicit examples for small s, an example showing that the operational and basic-row readings differ, structural consequences of the schedule's first rounds, and two obstructions that come from the input order.

8.1 Small sizes

An earlier computer search found ordered graphs realizing the schedule for every even $s \le 40$ (Appendix C). It extended one fixed ordered graph step by step, rerunning the whole execution after each proposed extension and rejecting any extension that changed an earlier round. Its program is not among the accompanying programs, but the inputs it found are. The program `search.py` (Appendix D) is a new search of the same kind. It finds other graphs, for every even $s \le 46$, so by the next proposition the schedule can be realized for every $s \le 47$. It starts from the normal form of Appendix C and designs one round at a time, in the manner of Knuth's matrices (answer 15(d)): at level L it splits the rows into groups of at most $L+1$, and for each group it adds the edges of one SAP that merges the group's rows, placed in the lists so that Algorithm H should find exactly that SAP. After each proposed group it reruns the whole execution from H1, and it rejects the group if an earlier round changes or the planned SAPs are not found. Neither search is guaranteed to succeed (`search.py` finds nothing for $s = 48$, 50, or 52 in two minutes each), and neither proves anything about larger s. We have replayed every example from both searches from H1 with two independent implementations of Algorithm H and checked its final matching with a separate maximum-matching program (Appendix D). A max-flow computation of ν in every round shows that all these examples are in fact basic-row realizations.

Proposition 8.1. *A realization for $s = 2h$ yields one for $s = 2h+1$: add an isolated girl–boy edge, numbered after all other vertices. The new realization is basic-row if the old one is.*

Proof. H1 matches the new edge first, no search ever reaches it, and $\lfloor 2h/2 \rfloor = \lfloor (2h+1)/2 \rfloor$, so the schedule is unchanged. □

8.2 Operational is weaker than basic-row

The matrices of answer 15(d) keep $\nu = f$ in every round (property F). The schedule does not require this. Take $s = 8$ and the ordered lists

$$[5,7];\ [3,5];\ [3];\ [1];\ [1,2];\ [3,4];\ [5,2,6];\ [7,8]. \tag{8.1}$$

The graph is a tree with 15 edges (Fig. 15). Step H1 leaves girls $4, 3, 2, 1$ free, and the run is

L	f	SAPs	W_5	W_{13}	W_{11}
2	4	2	15	7	8
3	2	1	11	3	5
4	1	1	7	4	4

which is the schedule $4 \to 2 \to 1 \to 0$. After the round at level 2 (round 2 of the run), the free girls are g_3, g_4 and the free boys b_6, b_8, and there are exactly three augmenting paths,

$$g_4 - b_1 = g_5 - b_2 = g_7 - b_6 \qquad \text{(level 3)},$$
$$g_3 - b_3 = g_2 - b_5 = g_1 - b_7 = g_8 - b_8 \qquad \text{(level 4)},$$
$$g_4 - b_1 = g_5 - b_2 = g_7 - b_5 = g_1 - b_7 = g_8 - b_8 \qquad \text{(level 5)}.$$

(In a tree there is at most one path between two vertices, so one only has to check the four pairs of a free girl and a free boy.) So with two free girls there is only one SAP, $\nu = 1 < f$, and property F fails for every choice of perfect matching. An operational realization need not be basic-row, and a search restricted to Knuth-style matrices could miss solutions.

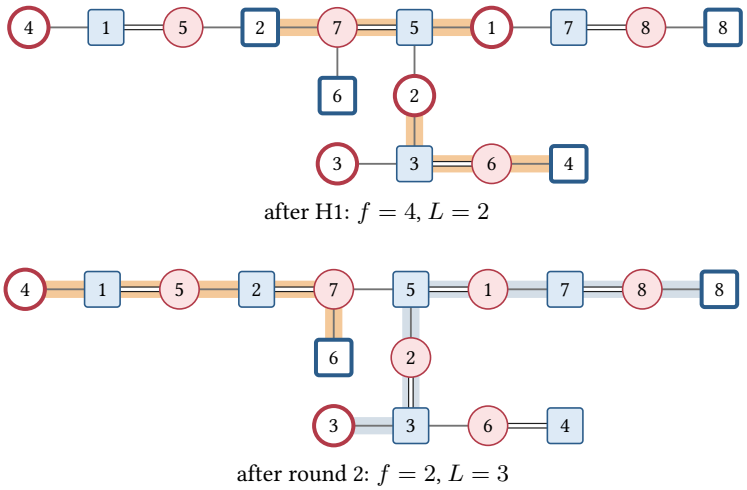


Figure 15. An operational realization for $s = 8$ that is not basic-row, labelled by the numbers of (8.1). Top: the matching after H1 and the two SAPs of the round at level 2. Bottom: the next matching. Its only SAP is shaded orange; the faint blue path is the augmenting path of level 4. (The third augmenting path, of level 5, is not drawn.)

8.3 The shape forced by the first matching

For even $s = 2h$ the schedule asks H1 to leave h girls free, so H1 must produce a maximal matching with only h couples, the fewest possible. The next proposition describes what that implies for the graph, whatever the input order. Write G° and $G^\bullet$ for the girls who are free and matched after H1, and $B^\bullet$ and B° for the boys who are matched and free after H1.

Proposition 8.2. *Suppose $s = M = N = 2h$ and* H1 *leaves h girls free. Then:*

(a) there are no edges between G° and B°, so the biadjacency matrix has the block form of Fig. 16;

(b) every perfect matching pairs G° with $B^\bullet$ and $G^\bullet$ with B°;

(c) whenever f girls are free later on, exactly f couples join $G^\bullet$ to $B^\bullet$, and $h - f$ couples are of each of the types G°–$B^\bullet$ and $G^\bullet$–B°;

(d) relative to any perfect matching, every basic path contains exactly one couple of type $G^\bullet$–$B^\bullet$.

Proof. (a) is maximality of the matching from H1. For (b), the h girls of G° can only be matched into the h boys of $B^\bullet$, which leaves B° for $G^\bullet$. (c) Vertices never become free again once matched, so $G^\bullet$ and $B^\bullet$ stay covered. If x, y, z count the current couples of types G°–$B^\bullet$, $G^\bullet$–$B^\bullet$, $G^\bullet$–B°, then $x + y = |B^\bullet| = h$, $y + z = |G^\bullet| = h$, and $x + y + z = 2h - f$; hence $y = f$. (d) A basic path starts at a free girl in G° and ends at a free boy in B°. Its perfect-matching edges stay within $G^\circ \cup B^\bullet$ or within $G^\bullet \cup B^\circ$ by (b), its couples of the two mixed types do too, and a couple of type $G^\bullet$–$B^\bullet$ is the only way to pass from the first part to the second; by (a) there is no way back. □

In terms of Knuth's matrices, each row consists of some indices of initially free girls followed by some indices of initially matched girls. For the 16-pair example this is the remark at the end of answer 15(d): an edge b_{even} — g_{odd} would join a free girl to a free boy of the first matching.

8.4 Realizations need cycles

The trees of Section 9 already force Algorithm H to do $\Theta(E\sqrt{V})$ work. No forest, however, can realize the schedule for large s: its first two rounds force many independent cycles. The argument

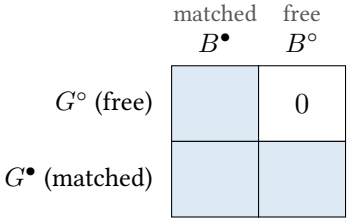


Figure 16. After a minimum-size H1 matching no free girl is adjacent to a free boy: the biadjacency matrix has a zero block.

uses only the prescribed numbers of SAPs in those rounds, so it applies to any algorithm that removes a maximal set of disjoint SAPs per round, in any order.

Theorem C. *Let an ordered graph with $s = M = N = 2h > 0$ realize the Tchoukaillon schedule. Let $a_2 = \lceil h/3 \rceil$ and $a_3 = \lceil (h - a_2)/4 \rceil$ be the numbers of SAPs in its rounds at levels 2 and 3, and let $p = 3a_2 - h \in \{0, 1, 2\}$. Then*

$$E \;\geq\; 3h + 4a_2 - 8p - a_3.$$

Consequently the cycle rank $\beta = E - V + \kappa$, where κ is the number of connected components, satisfies $\beta \geq a_2 - a_3 - 7p + 1 \geq \lceil (h - 76)/6 \rceil$. In particular the graph is not a forest if $h \geq 77$, or if $3 \mid h$; and for large s,

$$E \geq \tfrac{25}{12}\, s - O(1), \qquad \beta \geq \tfrac{1}{12}\, s - O(1).$$

Proof. Fix a perfect matching P and let M_1 be the matching produced by H1.

Basic paths after H1. The $2h$ endpoints of M_1 must touch all $2h$ edges of P, by maximality, so each edge of P has exactly one matched endpoint. Hence $P \cup M_1$ is a disjoint union of h basic paths of level 2, each of the form $u \text{ — } v \text{ ═ } w \text{ — } z$ with $u \in G^\circ$, $v \in B^\bullet$, $w \in G^\bullet$, $z \in B^\circ$; together they cover all vertices and use $3h$ edges.

Good triples. A SAP of the first dag round also has the form $u \text{ — } v \text{ ═ } w \text{ — } z$, and it meets at most three basic paths: the one containing u, the one containing the couple $v \text{ ═ } w$, and the one containing z. The round selects a_2 SAPs, and by maximality they meet all h basic paths. Call a selected SAP *good* if it meets three basic paths that no other selected SAP meets. If $q_1, \ldots, q_{a_2}$ are the numbers of basic paths met by the selected SAPs, put $\delta = 3a_2 - \sum q_i \geq 0$ and $\omega = \sum q_i - h \geq 0$, so that $\delta + \omega = p$. At most δ selected SAPs meet fewer than three paths; and a basic path met by $k \geq 2$ selected SAPs contributes $k - 1$ to ω and involves at most $k \leq 2(k - 1)$ of them. So the number g of good SAPs satisfies $g \geq a_2 - 2p$.

What a good triple becomes. Number the three basic paths met by a good SAP as $u_k \text{ — } v_k \text{ ═ } w_k \text{ — } z_k$ ($k = 1, 2, 3$), so that the SAP is $u_1 \text{ — } v_2 \text{ ═ } w_2 \text{ — } z_3$. After augmenting, the six pairs of the triple form two basic paths of level 3 (Fig. 17):

$$u_2 \text{ — } v_2 \text{ ═ } u_1 \text{ — } v_1 \text{ ═ } w_1 \text{ — } z_1 \qquad \text{and} \qquad u_3 \text{ — } v_3 \text{ ═ } w_3 \text{ — } z_3 \text{ ═ } w_2 \text{ — } z_2.$$

These are SAPs of the round at level 3, and their vertices lie in the dag at the levels given by their positions (a shorter route to one of them, followed by the rest of the path, would contain an augmenting path of level 2, and there is none after the first dag round). The triple contains $9 + 2 = 11$ edges of $P \cup M_1$ and of the selected SAP: the ten edges of the two new paths, and the old couple $v_2 w_2$. That edge joins the girl w_2, at position 3 of the second path, to the boy v_2, at position 1 of the first, so it points backwards and is not an arc of the level-3 dag.

The second round. Let F be the set of edges outside $P \cup M_1$ and outside the $2g$ selected edges of good triples; $|F| = E - 3h - 2g$. In the level-3 dag with F removed, each new path of a good triple is cut off from everything else: edges of $P \cup M_1$ never leave their triple, the selected edges stay

inside it, and the only remaining edge between the two new paths is not an arc. So a SAP that meets q of these $2g$ paths uses at least $q-1$ edges of F. The round at level 3 selects a_3 vertex-disjoint SAPs, which must meet all $2g$ new paths of good triples. Therefore $2g - a_3 \le |F| = E - 3h - 2g$, and $E \ge 3h + 4g - a_3 \ge 3h + 4a_2 - 8p - a_3$.

(When $h = 1$ there is no second round, $a_3 = 0$, and the bound is trivial.) The bound on β follows from $V = 4h$, $\kappa \ge 1$, and $h = 3a_2 - p$; the explicit form comes from substituting $a_3 = \lceil (2a_2 - p)/4 \rceil$ and checking $p = 0, 1, 2$, the weakest case being $p = 2$ with a_2 even. For $p = 0$ one gets $\beta \ge \lfloor a_2/2 \rfloor + 1 > 0$. The asymptotic statements follow from $a_2 = h/3 + O(1)$ and $a_3 = h/6 + O(1)$. □

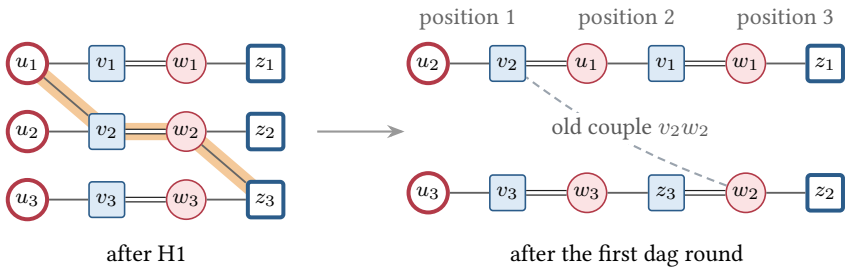


Figure 17. A good triple in the proof of Theorem C. Single lines are edges of the perfect matching P, double lines are couples. One SAP of the first dag round (shaded) meets three basic paths; afterwards their six pairs form two basic paths of level 3. The old couple v_2w_2 runs from position 3 back to position 1, so the level-3 dag cannot use it, and any SAP that visits both new paths must use some other edge.

Remark 8.3. For odd $s = 2h + 1$ the same method should give $E \ge 3h + 4a_2 - 8p - a_3 - 6$ and $\beta \ge \lceil (h - 124)/6 \rceil$, so that no forest realizes the schedule when $s \ge 251$. In outline: H1 now leaves h girls free, and either all h basic paths have level 2 and one couple of M_1 belongs to P, or one basic path has level 3 and the others level 2. We have not written out the second case in full (one must check that SAPs touching the long basic path do not spoil the isolation argument), so we state this only as a remark.

The case $p = 0$ also shows how property U fails in such a realization.

Corollary 8.4. *If a realization has $6 \mid s$, then just after its first dag round the SAPs have level 3 (five edges), and there is also an augmenting path with at least nine edges.*

Proof. With $p = 0$ all a_2 selected SAPs are good, so no basic path is left untouched, and the $2a_2$ new basic paths are disjoint SAPs of level 3, each meeting a selected SAP. Apply Proposition 7.1 with $q = 2a_2$ and $a = a_2$. □

8.5 The input order matters

One plan for the operational question is to find a graph with the right sequence of maximal SAP sets, and then look for an input order that makes Algorithm H pick those sets. The plan can fail, because Algorithm H does not choose its maximal sets freely: which SAP it finds depends on the order in QUEUE, on head insertion in step H6, and on the replacement rule of step H16.

A six-pair graph. Let the girls $g_1, \dots, g_6$ have the neighbour sets

$$\{b_1\};\quad \{b_1, b_2, b_5\};\quad \{b_3\};\quad \{b_2, b_3, b_4\};\quad \{b_3, b_5\};\quad \{b_5, b_6\}.$$

The edges $g_k b_k$ form a perfect matching. Starting from the matching $b_1 g_2$, $b_3 g_4$, $b_5 g_6$, the three augmentations

$$g_5 \text{ — } b_3 \text{ = } g_4 \text{ — } b_2, \qquad g_1 \text{ — } b_1 \text{ = } g_2 \text{ — } b_5 \text{ = } g_6 \text{ — } b_6,$$
$$g_3 \text{ — } b_3 \text{ = } g_5 \text{ — } b_5 \text{ = } g_2 \text{ — } b_2 \text{ = } g_4 \text{ — } b_4$$

are SAPs at levels 2, 3, 4, and after each of them no SAP of the same level remains. So the schedule for $s = 6$, which removes one SAP at each of these levels, is consistent with this graph.

Proposition 8.5. *No numbering and no list order make Algorithm H follow this schedule on this graph.*

Proof. A maximal matching with three couples must have exactly one endpoint on each edge $g_k b_k$ and contain none of them; the only such matching is $b_1 g_2$, $b_3 g_4$, $b_5 g_6$, so H1 must produce it, with free girls g_1, g_3, g_5. One SAP can be a maximal set only if it meets all three basic paths, and the only such SAP is g_5 — b_3 = g_4 — b_2. Before the first augmentation every root of the search succeeds, so this must be the SAP found from the first root, b_2, by taking the first arc of each list. Head insertion at b_3 then requires g_3 to precede g_5 in QUEUE, and head insertion at b_2 requires g_2 to precede g_4 at level 1. Girl g_2 is queued when g_1 discovers b_1, and g_4 when the earlier of g_3, g_5 discovers b_3; so g_1 comes first, and the queue must be (g_1, g_3, g_5). Step H16 then removes the last girl, g_5, leaving (g_1, g_3). In the next dag, level 1 is (g_2, g_5), both adjacent to b_5, so $\texttt{BLINK}[b_5] = (g_5, g_2)$ whatever the lists. Whichever of the free boys b_4, b_6 is searched first finds one of the basic paths g_1 — b_1 = g_2 — b_2 = g_4 — b_4 or g_3 — b_3 = g_5 — b_5 = g_6 — b_6; they are disjoint, so the other is found as well, and the round removes two SAPs instead of one. □

An exhaustive check of all $6!\,(3!)^2(2!)^2 = 103680$ girl numberings and list orders (the boy numbering plays no role) agrees: 740 of them give a three-couple H1 matching, and none of those follows the schedule. Other six-pair graphs do realize it (Appendix C).

Triangular chain graphs. Let Δ_{2h} have girls and boys $1, \ldots, 2h$, with g_j — b_i an edge whenever $i \le j$. Hegerfeld and Kratsch show that chain graphs of this kind can need many phases when the maximal sets are chosen adversarially [7, Section 5.3]. With Algorithm H and a minimum-size first matching, the order of the search rules out the Tchoukaillon schedule whenever $3 \mid h$:

Proposition 8.6. *Give Δ_{2h} any numbering and any list order. If* H1 *leaves h girls free and the first dag round finds a SAPs, then $3a > h$. So when $3 \mid h$, the graph Δ_{2h} cannot realize the schedule, whose first round would remove exactly $h/3$ SAPs.*

Proof. As in the previous proof, each edge $g_k b_k$ has exactly one matched endpoint. Let T be the set of k with g_k free (so b_k is matched) and S its complement (so b_k is free). Maximality forbids an edge from a free girl g_t to a free boy b_u; since $g_t b_u$ is an edge when $u \le t$, we must have $u > t$ for all $t \in T$ and $u \in S$. Hence $T = \{1, \ldots, h\}$ and $S = \{h+1, \ldots, 2h\}$. Write $u_i = g_i$, $v_i = b_i$, $x_j = g_{h+j}$, $z_j = b_{h+j}$ for $1 \le i, j \le h$; the matching from H1 is x_j = $v_{\pi(j)}$ for a permutation π, and the SAPs are exactly the paths

$$u_i \text{ — } v_{\pi(j)} \text{ = } x_j \text{ — } z_d \qquad \text{with } \pi(j) \le i \text{ and } d \le j.$$

Suppose $a < h$ SAPs are selected, let i_0 be the largest index of an unselected free girl and d_0 the smallest index of an unselected free boy. An unselected x_j with $\pi(j) \le i_0$ and $j \ge d_0$ would give a SAP disjoint from the selected ones; so each of the $h - a$ unselected x_j has $\pi(j) > i_0$ or $j < d_0$. At most $h - i_0 \le a$ of them satisfy the first condition and at most $d_0 - 1 \le a$ the second; hence $h \le 3a$.

Suppose $h = 3a$. Then all these inequalities are equalities: the selected free girls are $u_{2a+1}, \dots, u_{3a}$, the selected free boys are $z_1, \dots, z_a$, every selected x_j has $\pi(j) \le 2a$ and $j > a$, and every x_j with $\pi(j) > 2a$ is unselected and has $j > a$. Girl u_{2a} is never selected. Each selected SAP passes through a boy $v_{\pi(j)}$ with $\pi(j) \le 2a$, whose list contains u_{2a}; that arc cannot have been consumed earlier, because following it would have produced a SAP ending at u_{2a}. Since lists are in reverse queue order, every selected free girl comes after u_{2a} in QUEUE. Only the selected free girls are adjacent to the boys $v_{2a+1}, \dots, v_{3a}$, while u_{2a} is adjacent to all of $v_1, \dots, v_{2a}$; so the latter are all discovered first, and their mates come first in level 1. In particular every x_j with $\pi(j) > 2a$ is queued after every selected x_j. Now consider the first root of the search. Before any augmentation every root succeeds, so it is a selected free boy z_d with $d \le a$. All x_j with $\pi(j) > 2a$ are adjacent to it (since $j > a \ge d$) and come later in the queue, so the head of BLINK$[z_d]$ is not a selected girl; and the search from it succeeds. So the first SAP found uses an x_j that should have remained unselected, a contradiction. □

For $h = 3$ the schedule is consistent with the graph itself: starting from the matching x_1v_1, x_2v_3, x_3v_2, the graph Δ_6 admits the single maximal SAPs

$$u_3 — v_2 ═ x_3 — z_1, \qquad u_1 — v_1 ═ x_1 — z_1 ═ x_3 — z_2,$$
$$u_2 — v_2 ═ u_3 — v_3 ═ x_2 — z_2 ═ x_3 — z_3$$

at levels $2, 3, 4$. It is the order of Algorithm H's search that rules them out. Other graphs with $s = 6$ do realize the schedule.

8.6 What an induction would need

One might try to pass from $s = 2h$ to $s = 2h + 2$ by adding two pairs. The schedule changes in a very controlled way. Write $f_L(h)$ and $a_L(h)$ for the schedule with $f_2(h) = h$, extended by zeros.

Lemma 8.7. *Let K be the first level $L \ge 2$ such that $L+1$ divides $f_L(h)$. Then $f_L(h+1) = f_L(h)+1$ for $2 \le L \le K$ and $f_L(h+1) = f_L(h)$ for $L > K$; so $a_L(h+1) = a_L(h) + [L = K]$.*

Proof. The map $x \mapsto x - \lceil x/(L+1) \rceil$ increases by 1 from x to $x+1$, except when $(L+1) \mid x$, where it stays the same. Start from a difference of 1 at level 2 and follow it. □

So adding two pairs should add one free girl, keep every count of SAPs except one extra at level K, and then rejoin the old schedule. Doing this with a fixed set of edges, while preserving every earlier distance and every earlier BLINK order, is what we have not managed: an edge added for a later round can create a shorter path in an earlier one, or change the order in which an earlier search discovers vertices.

9 Simpler examples for the running time

If one only wants inputs on which Algorithm H does $\Theta(E\sqrt{V})$ work, without the exact maximum number of rounds, a much simpler family will do. It is the folklore example of disjoint paths with $3, 5, 7, \dots$ edges, on which every phase is forced (compare Hegerfeld and Kratsch's paths [7, Section 5.1]), here joined into a tree and given an input order for Algorithm H, with exact operation counts.

For $k = 2, 3, \dots, K$ let $c_k = k(k-1)/2$ and form a path on the girls $x_i = g_{c_k+i}$ and boys $y_i = b_{c_k+i}$, $0 \le i < k$:

$$x_0 — y_0 — x_1 — y_1 — \cdots — x_{k-1} — y_{k-1}, \qquad \text{with lists} \quad x_0 \colon [y_0], \quad x_i \colon [y_{i-1}, y_i].$$

Then, for each $k < K$, append the first boy of component $k+1$ to the end of the list of the last girl x_{k-1} of component k. The result is a tree (in fact a caterpillar) of maximum degree three with

$$M = N = s = \frac{K(K+1)}{2} - 1, \qquad E = 2s - 1.$$

For $K = 4$, the case of Fig. 18, the lists are

$$[1];\ [1,2,3];\ [3];\ [3,4];\ [4,5,6];\ [6];\ [6,7];\ [7,8];\ [8,9].$$

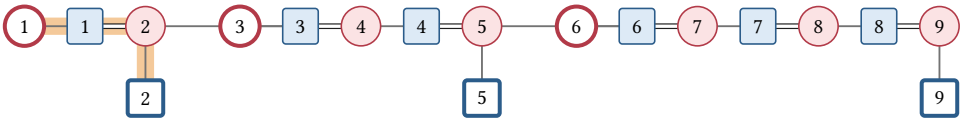


Figure 18. The tree for $K = 4$ after H1, with the SAP of the first dag round. Component k is a path whose unique augmenting path has level k; the thin links between components never enter a dag.

Proposition 9.1. *On this tree Algorithm H performs K rounds. For $j = 2, \ldots, K$ the round at level $L = j$ finds one SAP, which completes component j, and*

$$W_5 = (2j-1)(K-j+1) + [j < K], \qquad W_{13} = j, \qquad W_{11} = j(K-j+1).$$

In total $\sum W_5 = K(K-1)(2K+5)/6 + K - 2$, $\sum W_{13} = K(K+1)/2 - 1$, and $\sum W_{11} = K(K-1)(K+4)/6$, while H1 *examines s entries. Hence $\sum W_5/(E\sqrt{V}) \to 1/3$.*

Proof. Step H1 scans each component's girls $x_{k-1}, \ldots, x_1$ before x_0 and couples $x_i == y_{i-1}$; it never reaches a link. So each component has a single free girl x_0 and a single free boy y_{k-1}, its unique augmenting path has level k, and QUEUE lists the free girls from component K down to component 2. The round at level j completes component j, whose free girl is last in the queue, so step H16 preserves the order.

The links do not interfere. A link goes from a girl of component k to the first boy of component $k+1$, so an alternating path, whose steps from girls to boys follow edges, can only move to components of higher index. In round j the free girls lie in components $j, \ldots, K$, so the completed components $2, \ldots, j-1$ are never entered. The link out of component k starts at the girl x_{k-1}, which lies at level $k-1$; in round j only levels up to $j-1$ are scanned, so this link is examined only in round k, and then its boy is already in B_1, so step H6 ignores it. Hence the dags are exactly those of the separate paths.

In round j each of the $K-j+1$ unresolved components contributes its girls at levels $0, \ldots, j-1$: one girl with one neighbour and $j-1$ with two, $2j-1$ examinations in all, plus one for the link of component j if $j < K$. The shortest component is scanned last at every level, so the boys at levels $1, \ldots, j$ of all unresolved components are marked before the free boy is found: $W_{11} = j(K-j+1)$. The only root is the free boy of component j, and its search consumes j arcs. □

Here $\sum W_{13} = \Theta(E)$ but $\sum W_{11} = \Theta(E\sqrt{V})$: the mark-resetting loop of step H11 is as expensive as the breadth-first search, and cannot be left out of a running-time analysis.

Remark 9.2. Replacing component k by a cycle C_{4k-2}, with suitable lists, gives a 2-regular graph with $s = K^2 - 1$ on which every round finds one SAP and $\sum W_5/(E\sqrt{V}) \to 1/(3\sqrt{2})$. The staircase graphs do better on both counts: they have the largest possible number of rounds, about $2\sqrt{s}$, against about $\sqrt{2s}$ for the tree and $\sqrt{s}$ for the cycles, and $\sum W_5/(E\sqrt{V}) \to 2\sqrt{2}/3$ (Theorem W).

10 Credits, literature, and open problems

10.1 Where the ideas come from

Augmenting paths and phases. The union of two matchings and the $s - s'$ disjoint augmenting paths are due to Berge [1]. Hopcroft and Karp [8] introduced phases of vertex-disjoint shortest augmenting paths. They proved that path lengths never decrease and that paths of one length found in sequence are vertex-disjoint (their Theorem 2 and Corollaries 3 and 4), and bounded the number of distinct lengths by $2\lfloor\sqrt{s}\rfloor + 2$ (their Theorem 3), which gives running time $O((V + E)\sqrt{s})$ in our notation. The paper contains no lower-bound example. Karzanov obtained the same bound independently [10]. As the pre-fascicle notes on page 7, Dinitz's shortest-augmenting-path algorithm for network flow, with improvements by Even and Itai, is essentially the same procedure when specialized to bipartite matching [4]; see also Even and Tarjan [6]. On random graphs long SAPs are rare, so the algorithm typically needs far fewer rounds [14, 2]. Knuth's Algorithm H, its data structures, Theorem H, Corollary K, and exercises 5, 15, and 16 are in [11]; Lemma 3.1 is exercise 15(a,b) with Knuth's answer spelled out.

Tchoukaillon numbers. The sieve of exercise 12 and the numbers Ч_n go back to Erdős and Jabotinsky [5], who studied their growth; Broline and Loeb [3] connected them with the Tchoukaillon game and its $1/\pi$ asymptotics (Knuth's answer to exercise 11 corrects an error bound claimed there); see also Jones, Taalman, and Tongen [9] and OEIS sequence A002491 [15]. Exercise 11(d) gives $\text{Ч}_n = n^2/\pi + O(n^{4/3})$, and Li [13] has extended such asymptotics to the whole Tchoukaillon array. Knuth's 2022 lecture [12] linked Tchoukaillon numbers to the question whether the Hopcroft–Karp algorithm can ever take nonlinear time.

Lower bounds on phases. Hegerfeld and Kratsch [7] study algorithms that follow the Hopcroft–Karp phase framework, and show that such an algorithm *may* need $\Omega(\sqrt{n})$ phases on paths, on trivially perfect graphs, and on chain graphs, when the maximal set of SAPs removed in each phase is chosen adversarially. Their constructions do not address whether a particular implementation, with its own data structures, can be steered into those choices by its input; Section 8.5 shows that this can fail. On graphs whose components have unique augmenting paths, such as the paths of Section 9, every phase is forced, and examples of this kind are folklore. We found no earlier statement of the exact value of $R(s)$.

New here. As far as we know, the following are new: the exact value of $R(s)$ and the staircase graphs (Theorems R, S, and W); the comparison $Q_n \le \text{Ч}_n$; the halving theorem and the classification of the literal reading (Theorem U, Corollary L); and the results of Section 8. The upper bound in Theorem R is a direct refinement of the counting in Theorem H and in [8]. We would be grateful to hear of earlier sources for any of these results.

10.2 Open problems

1. **The operational question.** Does every s admit an ordered graph realizing the Tchoukaillon schedule? Even an infinite family of sizes would be progress. By Theorem C such a family must have at least $s/12 - O(1)$ independent cycles.
2. **The basic-row question.** The same, keeping f disjoint SAPs in every round, as in Knuth's matrices. A positive answer implies one to the first question; by Section 8.2 a negative answer would not settle it. All computed examples (Appendix C) are basic-row.

3. **Most rounds from the largest start.** What is the largest number of rounds if H1 must leave $\lfloor s/2 \rfloor$ girls free? It is at most $R(s)$, and at least $T(s)$ for every s at which the schedule can be realized; the staircase graphs start with only about $2\sqrt{s}$ free girls.
4. **Few edges.** How many edges does an operational realization need? Theorem C gives $E \geq \frac{25}{12}s - O(1)$, using only two rounds.
5. **An explicit threshold.** Make explicit the size beyond which $R(s) > T(s)$ always holds; Section 6 confirms it for $64 \leq s < 3.2 \times 10^9$.

A An input order for Knuth's sixteen-pair graph

Answer 15(d) gives the graph but not an input order. Here is one under which Algorithm H follows the Tchoukaillon schedule and passes through Knuth's matrices. We keep Knuth's logical names g_k, b_k. The edges are $g_k b_k$ for $1 \leq k \leq 16$ and

$$\begin{gathered} b_1g_2,\ b_3g_4,\ b_5g_6,\ \ldots,\ b_{15}g_{16}, \\ b_2g_4,\ b_3g_5,\ b_8g_{10},\ b_9g_{11},\ b_{14}g_{16},\ b_4g_6,\ b_5g_8,\ b_7g_9,\ b_6g_{10},\ b_3g_7,\ b_{10}g_{12}, \end{gathered} \tag{A.1}$$

so $M = N = s = 16$ and $E = 35$.

The input. Throughout this appendix g_k and b_k are Knuth's logical names; the numbers that Algorithm H sees as girl indices are a separate thing, given in the column "number" below. Boys keep their numbers. Girls receive the numbers $16, 15, \ldots, 1$ in the order

$$g_{16}, g_{14}, g_{12}, g_{10}, g_8, g_6, g_4, g_2,\ g_1, g_3, g_7, g_{13}, g_9, g_{15}, g_{11}, g_5.$$

Each girl's list is in increasing order of boys, except that an even girl g_k puts b_{k-1} first. In numerical form:

number	girl	list	number	girl	list
1	g_5	$[3,5]$	9	g_2	$[1,2]$
2	g_{11}	$[9,11]$	10	g_4	$[3,2,4]$
3	g_{15}	$[15]$	11	g_6	$[5,4,6]$
4	g_9	$[7,9]$	12	g_8	$[7,5,8]$
5	g_{13}	$[13]$	13	g_{10}	$[9,6,8,10]$
6	g_7	$[3,7]$	14	g_{12}	$[11,10,12]$
7	g_3	$[3]$	15	g_{14}	$[13,14]$
8	g_1	$[1]$	16	g_{16}	$[15,14,16]$

The run. Step H1 scans the even girls first; each takes b_{k-1}, so the matching is b_1g_2, b_3g_4, …, $b_{15}g_{16}$, as in Knuth's first matrix, and H1 examines 20 entries. The rounds are:

L	f	SAPs	W_5	W_{13}	W_{11}	SAPs found, as new couples (g, b)
2	8	3	35	16	16	$(5,3), (4,2)$; $(11,9), (10,8)$; $(15,15), (16,14)$
3	5	2	34	14	15	$(9,7), (8,5), (6,4)$; $(13,13), (14,14), (16,16)$
4	3	1	28	10	12	$(7,3), (5,5), (8,8), (10,6)$
5	2	1	25	6	12	$(3,3), (7,7), (9,9), (11,11), (12,10)$
6	1	1	24	6	12	$(1,1), (2,2), (4,4), (6,6), (10,10), (12,12)$

Here girls have their logical names. The free girls before the rounds are, in `QUEUE` order, $(1,3,7,13,9,15,11,5)$, $(1,3,7,13,9)$, $(1,3,7)$, $(1,3)$, (1); every SAP ends at the last free girl, so step H16 never reorders the queue. After each round the basic paths with respect to the matching $g_k b_k$ are exactly the rows of the next matrix in Fig. 5, and the run ends with that perfect matching.

All augmenting paths. Enumerating every simple augmenting path before each round gives the following counts, by level.

before the round at	numbers of augmenting paths of each level
$L = 2$	22 of level 2, 4 of level 3
$L = 3$	16 of level 3, 6 of level 5
$L = 4$	7 of level 4, 12 of level 6, 8 of level 8
$L = 5$	3 of level 5, 4 of level 7, 4 of level 9, 2 of level 11
$L = 6$	1 of level 6

So property U fails in every round but the last, while property F holds throughout. The paths π_2 and π_3 of (3.3) are two of the 26 counted in the first line.

B The halving graphs

These graphs show that Theorem U is sharp and supply the literal examples in Corollary L. Let $f = 2^t$ with $t \geq 0$. For a positive integer i let $\rho(i)$ be the exponent of the largest power of 2 that divides i (the "ruler function"; for example $\rho(12) = 2$). The graph has girls u_i, c_i and boys b_i, d_i for $1 \leq i \leq f$, and the edges

$$u_i b_i, \quad c_i b_i, \quad c_i d_i \quad (1 \leq i \leq f), \qquad u_i\, b_{i+2^{\rho(i)}} \quad (1 \leq i < f).$$

So $M = N = s = 2f$ and $E = 4f - 1$; the edges $u_i b_i$ and $c_i d_i$ form a perfect matching.

The input. Number b_i as $2i - 1$ and d_i as $2i$, and number c_i as $f + i$, with list $[b_i, d_i]$. Sort the indices i by decreasing $\rho(i)$, breaking ties by increasing i, and give the corresponding girls u_i the numbers $f, f - 1, \ldots, 1$ in that order. The list of u_i is $[b_i, b_{i+2^{\rho(i)}}]$, or $[b_f]$ when $i = f$. For the three literal examples of Corollary L this gives

s	girl lists, in increasing order of girls
2	$[1]$; $[1, 2]$
4	$[1, 3]$; $[3]$; $[1, 2]$; $[3, 4]$
8	$[5, 7]$; $[1, 3]$; $[3, 7]$; $[7]$; $[1, 2]$; $[3, 4]$; $[5, 6]$; $[7, 8]$

The run. Step H1 couples every c_i with b_i and leaves all u_i free, in the order just described. After r dag rounds ($0 \leq r \leq t$) the free girls are the u_i with $2^r \mid i$, and the indices fall into *blocks* of 2^r consecutive integers $i - 2^r < j \leq i$, one block for each free girl u_i. Her basic path visits $r + 1$ of the girls u_j of her block, namely u_i and then u_j for $j = i - 2^r + 2^q$ with $q = r - 1, \ldots, 1, 0$, and ends with c — d at the smallest index of the block; it has level $r + 2$. For example, when $f = 4$ and $r = 1$ the free girls are u_2 and u_4, and the basic path of u_4 is u_4 — b_4 = u_3 — b_3 = c_3 — d_3.

Every augmenting path has level $r + 2$ at this stage: a free girl can start along b_i or along $b_{i+2^{\rho(i)}}$, and after that first step the alternating continuation is forced. In the next round step H6 lists, at each b_i, the free girls in reverse queue order. As a result the search joins each block whose free girl u_i has $\rho(i) = r$ to the block on its right; the other roots fail at deleted girls; and step H16 removes exactly the girls with $\rho(i) = r$, which form the end of the queue. So the free girls go $f \to f/2 \to \cdots \to 1 \to 0$ in $t + 1$ dag rounds, and property U holds in every round. (For $f \leq 32$ we have also checked this by simulation, including a census of all augmenting paths.)

Work. The round that starts with $n' = f/2^r$ free girls has $W_5 = 2n'(r + 2) - 1$ and $W_{13} = W_{11} = n'(r + 2)$; in total $\sum W_5 = 12f - 3t - 9$ and $\sum W_{13} = \sum W_{11} = 6f - t - 4$, and H1 examines $3f - 1$ entries. So these graphs need only linear work in all.

C Computed realizations for small sizes

For every even $s = 2h \le 40$ the earlier search described in Section 8.1 found an ordered graph that realizes the Tchoukaillon schedule; by Proposition 8.1 this covers all $s \le 41$. In each example H1 produces the matching $g_{h+j} == b_{2j-1}$ $(1 \le j \le h)$ and leaves girls $h, h-1, \ldots, 1$ free, in that order. The table gives the number of edges, the number of rounds (counting H1), the work counts summed over all rounds after H1, and the numbers of SAPs removed at levels $2, 3, \ldots$. The entry for $s = 16$ is Knuth's graph with the input of Appendix A. The complete ordered lists are in the file `code/data/numerical_witnesses.json`, and `python algorithm_h.py catalog` s `--json` prints every queue, dag, and SAP of the run on any of them (Appendix D).

s	E	rounds	H1	W_5	W_{13}	W_{11}	SAPs removed
2	3	2	2	3	2	2	1
4	7	3	5	12	7	7	1, 1
6	12	4	7	34	14	18	1, 1, 1
8	15	4	9	37	16	20	2, 1, 1
10	21	5	13	67	29	33	2, 1, 1, 1
12	28	6	16	133	46	60	2, 1, 1, 1, 1
14	31	6	19	123	49	56	3, 1, 1, 1, 1
16	35	6	20	146	52	67	3, 2, 1, 1, 1
18	43	7	25	212	77	93	3, 2, 1, 1, 1, 1
20	46	7	27	223	78	97	4, 2, 1, 1, 1, 1
22	55	8	30	367	109	148	4, 2, 1, 1, 1, 1, 1
24	60	8	34	378	117	156	4, 2, 2, 1, 1, 1, 1
26	63	8	39	362	118	149	5, 2, 2, 1, 1, 1, 1
28	67	8	39	363	121	150	5, 3, 2, 1, 1, 1, 1
30	77	9	45	515	164	208	5, 3, 2, 1, 1, 1, 1, 1
32	79	9	47	542	164	221	6, 3, 2, 1, 1, 1, 1, 1
34	91	10	55	709	214	270	6, 3, 2, 1, 1, 1, 1, 1, 1
36	97	10	56	748	224	285	6, 3, 2, 2, 1, 1, 1, 1, 1
38	100	10	58	779	220	299	7, 3, 2, 2, 1, 1, 1, 1, 1
40	104	10	61	716	225	272	7, 4, 2, 2, 1, 1, 1, 1, 1

The program `search.py` of Section 8.1 finds different realizations for every even $s \le 46$. They have the same matching after H1 and the same order of the free girls, and their ordered lists are in the file `code/data/search_results.json`. Those beyond the table above are

s	E	rounds	H1	W_5	W_{13}	W_{11}	SAPs removed
42	121	11	65	1138	290	394	7, 4, 2, 2, 1, 1, 1, 1, 1, 1
44	121	11	67	1092	290	393	8, 4, 2, 2, 1, 1, 1, 1, 1, 1
46	127	11	71	1123	298	408	8, 4, 3, 2, 1, 1, 1, 1, 1, 1

and by Proposition 8.1 they cover every $s \le 47$.

D Programs and checks

The proofs in these notes do not depend on computation. The programs make the examples checkable and catch mistakes. They are in the directory `code/` of the repository github.com/boonsuan/hopcroftkarp, and need only Python 3.10 or later and its standard library (NumPy is used only by the longest check).

Simulators. `algorithm_h.py` is a literal transcription of Algorithm H: it builds the edge nodes as in answer 6, runs step H1 as in answer 7, and runs steps H2–H17 under an explicit program counter. It records every queue, `MARKED` order, `BLINK` list, SAP, and the counts W_5, W_{13}, W_{11}. `independent_audit.py` implements the same rules with ordinary loops and lists and shares no code with it; the two must produce identical records. `augmenting.py` contains computations that do not use Algorithm H at all: maximum matchings by plain augmenting paths, an exhaustive list of augmenting paths for small graphs, and ν by a max-flow computation.

Examples and figures. `examples.py` builds every named graph of these notes (the staircase graphs Γ_n and their padding to any s, Knuth's 16-pair graph with the input of Appendix A, the 8-pair tree, the halving, caterpillar, six-pair, and triangular chain graphs, and the catalog of Appendix C), together with the schedule and the numbers $T(s)$, $R(s)$, Q_n, Ψ_n. `make_figures.py` draws every figure that shows a matching, dag, or SAP of a specific graph by running the simulator, so the pictures show exactly what Algorithm H does.

Search. `search.py` is the search of Section 8.1. For each even s it builds an ordered graph one round of the Tchoukaillon schedule at a time, rerunning Algorithm H from H1 after every proposed change, and replays what it finds with both simulators. Its random choices are seeded, so every run finds the same graphs; they are written to `code/data/search_results.json`. The module's documentation describes the method in full.

Checks. `verify.py` runs fifteen named checks, one for each group of computed statements in these notes, and prints a pass/fail line for each. They include the hook description of Lemma 5.3 and the search of Lemmas 5.5 and 5.6 for $n \le 60$, the count $2^m - 1$ of SAPs (by exhaustive enumeration for $n \le 9$), the counts of Theorem W, the padding to every size, the claim about the corner edge in Section 5.1, the values of Ψ_n and the comparison with Q_n, the replay of every example and table, $\nu = f$ in the catalog and in the graphs found by `search.py`, the tree of Section 8.2, and the six-pair and chain obstructions.

Reproducing. From `code/`:

`python verify.py`	all checks, about 11 seconds
`python verify.py --full`	full ranges, including Ψ_n for $n \le 10^5$ and all 103680 six-pair inputs (about 40 seconds)
`python make_figures.py`	redraws `report/figures/gen_*`
`python search.py 46`	searches every even $s \le 46$ (about two minutes)
`python algorithm_h.py gamma 5`	prints a run (try `--help`)

and from `report/`, two runs of `pdflatex main.tex` build these notes. The website hopcroftkarp.site has its own port of the simulator, tested step by step against traces exported from `algorithm_h.py`.

References

[1] Claude Berge, "Two theorems in graph theory," *Proceedings of the National Academy of Sciences of the USA* **43** (1957), 842–844. doi:10.1073/pnas.43.9.842.

[2] Holger Bast, Kurt Mehlhorn, Guido Schäfer, and Hisao Tamaki, "Matching algorithms are fast in sparse random graphs," *Theory of Computing Systems* **39** (2006), 3–14.

[3] Duane M. Broline and Daniel E. Loeb, "The combinatorics of Mancala-type games: Ayo, Tchoukaillon, and $1/\pi$," *UMAP Journal* **16** (1995), 21–36. arXiv:math/9502225.

[4] Yefim Dinitz, "Dinitz' algorithm: The original version and Even's version," in *Theoretical Computer Science: Essays in Memory of Shimon Even*, Lecture Notes in Computer Science **3895** (2006), 218–240. doi:10.1007/11685654_10.

[5] Paul Erdős and Eri Jabotinsky, "On sequences of integers generated by a sieving process," *Indagationes Mathematicae* **20** (1958), 115–128.

[6] Shimon Even and Robert Endre Tarjan, "Network flow and testing graph connectivity," *SIAM Journal on Computing* **4** (1975), 507–518.

[7] Falko Hegerfeld and Stefan Kratsch, "On adaptive algorithms for maximum matching," *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, LIPIcs **132** (2019), 71:1–71:16. doi:10.4230/LIPIcs.ICALP.2019.71; arXiv:1904.11244.

[8] John E. Hopcroft and Richard M. Karp, "An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs," *SIAM Journal on Computing* **2** (1973), 225–231. doi:10.1137/0202019.

[9] Brant Jones, Laura Taalman, and Anthony Tongen, "Solitaire mancala games and the Chinese remainder theorem," *American Mathematical Monthly* **120** (2013), 706–724. arXiv:1112.3593.

[10] A. V. Karzanov, "An exact estimate of an algorithm for finding a maximum flow, applied to the problem on representatives" (in Russian), in *Problems in Cybernetics: Proceedings of the Seminar on Combinatorial Mathematics*, Moscow, 1973, 66–70.

[11] Donald E. Knuth, *The Art of Computer Programming*, Volume 4, Pre-Fascicle 14A: *Bipartite Matching*, Section 7.5.1. Draft of 18 September 2026 (zeroth printing, revision −88). Page numbers in these notes refer to this draft.

[12] Donald E. Knuth, "Tchoukaillon numbers," Rutgers Experimental Mathematics Seminar, 27 January 2022. Abstract at dimacs.rutgers.edu; video at vimeo.com/672468553.

[13] Shisheng Li, "Asymptotics of the Tchoukaillon array and a conjecture of Beluhov," 2026. arXiv:2608.17517.

[14] Rajeev Motwani, "Average-case analysis of algorithms for matchings and related problems," *Journal of the ACM* **41** (1994), 1329–1356.

[15] OEIS Foundation, *The On-Line Encyclopedia of Integer Sequences*, sequence A002491. oeis.org/A002491.