Superpermutations

Notes on the lower bound

Version 0.1.1

Robin Houston

November 26, 2018

Abstract

Some ideas towards improving the lower bound $n! + (n-1)! + (n-2)! + (n-3)$ for the length of a superpermutation on n symbols.

Let $n \geq 3$ be an integer,[1] and let $N = \{1, \ldots, n\}$.

1 General definitions

This section is ludicrously terse, since its purpose is simply to fix particular terminology and notation for familiar concepts.

1.1 From words to superpermutations

A **word** is a finite – possibly empty – sequence of elements of N. The set of all words is denoted N^*. Concatenation of words is denoted by juxtaposition. The **length** of a word u is the number of elements it contains, denoted by $|u|$. By abuse of notation, we identify each $c \in N$ with the corresponding word of length 1. If $u = xyz$ for words u, x, y, z, we say that x is a **prefix** of u, y is a **factor** of u, and z is a **suffix** of u. If furthermore $|x| < |u|$, we say x is a **proper prefix** of u. Let $u[i, j)$ denote the (unique, if it exists) word b such that $u = xyz$ for words x, y, z with $|x| = i$ and $|xy| = j$. (If $0 \leq i \leq j \leq |u|$ for integers i, j then the word $u[i, j)$ exists and has length $(j - i)$. Every factor of u can be written in this way, since if $u = xyz$ then $y = u[|x|, |xy|)$.)

Two words u and v are **cyclically equivalent**, denoted $u \sim v$, just when there exist words x and y such that $u = xy$ and $v = yx$. The **cycle class** of a word u, denoted $[u]$, is the set $\{v \in N^* \mid v \sim u\}$.

Lemma 1. *If $u \sim v$, and u (hence v) contains each element of N at most once, then:*

- *If $u = v$ there are two pairs (x, y) such that $u = xy$ and $v = yx$;*
- *If $u \neq v$ there is one pair (x, y) such that $u = xy$ and $v = yx$.*

[1]The only reason to exclude the cases $n = 1$ and $n = 2$ globally is that it avoids the need to exclude them specifically later on: 1-cycles only exist for $n > 1$, and 2-cycles only exist for $n > 2$.

Proof. Suppose there are two different pairs (x, y) and (x', y'), with $u = xy = x'y'$ and $v = yx = y'x'$. We wish to show that this can happen in only one way, and only if $u = v$.

If $x = x'$ then $y = y'$ and vice versa; so, since the pairs are different, we must have $x \neq x'$ and $y \neq y'$.

Since x and x' are both prefixes of u, one must be a prefix of the other. Suppose, without loss of generality by symmetry, that x is a prefix of x'. It follows – since y and y' are both prefixes of v with $|y'| < |y|$ – that y' is a prefix of y.

If $|x| > 0$ then x has a first element, which is also the first element of x'. This element must occur in v at two different positions, since $v = yx = y'x'$ and $y \neq y'$, contradicting the assumption that v contains each element at most once.

Similarly, if $|y'| > 0$ then y' and y have a common first element, which must occur in u at two different places.

So the only way for there to be two different pairs (x, y) and (x', y') is for x and y' to both be empty. It follows that $y = x' = u = v$, which completely determines the two pairs and is possible only if $u = v$. □

1.2 Directed and undirected graphs

Directed graphs: A (weighted) **digraph** G consists of a set $\mathrm{nodes}(G)$ of **nodes** and a set $\mathrm{edges}(G)$ of **edges**. Each edge $e \in \mathrm{edges}(G)$ has a **source** $\mathrm{source}(e) \in \mathrm{nodes}(G)$, a **target** $\mathrm{target}(e) \in \mathrm{nodes}(G)$, and a **weight** $w(e) \in \mathbb{Z}_{\geq 0}$. A **walk** is a sequence alternating between nodes and edges, beginning and ending with a node, in which each edge is preceded by its source and followed by its target. A **path** is a walk all of whose nodes are distinct. A **cycle** is a walk whose first and last nodes are the same, and whose nodes are otherwise distinct. The **length** $|p|$ of a walk p is the number of edges it contains; the **weight** $w(p)$ of a walk p is the sum of the weights of those edges. A **Hamiltonian path** is a path that visits all nodes. A digraph is **complete** if for every pair x, y of nodes there is a unique edge with source x and target y. We denote this edge $x \to y$.

Undirected graphs: a **graph** H consists of a set $\mathrm{nodes}(H)$ of **nodes** and a symmetric reflexive **adjacency relation** $\sim_H \subseteq \mathrm{nodes}(H) \times \mathrm{nodes}(H)$. We say there is an **edge** between nodes x and y if $x \neq y$ and $x \sim_H y$. The **connected components** of H are the equivalence classes of the transitive closure of $\sim_H$.

2 Superpermutations and the permutation digraph

2.1 Permutations and superpermutations

A **permutation** is a word that contains each element of N precisely once – and so has length n. The set of all permutations is denoted P_n, and $|P_n| = n!$. A **superpermutation** is a word that has every permutation as a factor.

2.2 Overlap distance and the permutation digraph

Given two words u and v, the **overlap distance** $d(u, v)$ is defined to be the minimum value of $|z|$ over all triples of words (x, y, z) for which $xy = u$ and $yz = v$. Observe that $0 \leq d(u, v) \leq |v|$, that $d(u, u) = 0$, that $d(u, w) \leq d(u, v) + d(v, w)$, and that $d(u, v) \neq d(v, u)$ in general.

Let the **permutation digraph** G_n be the complete digraph with nodes$(G_n) = P_n$, and $w(u \to v) = d(u, v)$.

2.3 Using G_n to find lower bounds

We are interested in lower bounds on the length of a superpermutation s. We shall show how to convert a superpermutation to a Hamiltonian path in G_n, which allows us to transfer a lower bound on the weight of a Hamiltonian path p in G_n to a lower bound on $|s|$, and also justifies us in considering a restricted family of such paths.

Definition 2.1. An edge $u \to v$ in G_n is **improper** if there is a permutation w different from both u and v such that $w(u \to w \to v) = w(u \to v)$. Otherwise it is **proper**.

An improper edge $u \to v$ that decomposes as $u \to w \to v$ in this way is said to **skip over** the permutation w.

Example 2.1. For example, in G_4 the edge $1234 \to 3412$ of weight 2 is improper, because it decomposes as

$$1234 \to 2341 \to 3412,$$

which consists of two edges of weight 1.

Definition 2.2. A path in G_n is **well-behaved** if, whenever it has an improper edge $u \to v$ that skips over some permutation w, the path does not visit w before it visits u.

Lemma 2. *Given a superpermutation s, we may construct a well-behaved Hamiltonian path p in G_n such that $w(p) \leq |s| - n$. The path p has the property that, for any edge $e : u \to v$ in the path, if there is any path $u \to v$ whose weight is*

Proof. Let s be a superpermutation. For each permutation u, let aub be the unique factorisation of s such that u is not a factor of b. In other words, if u occurs more than once in s we choose its last occurrence. Let $i_q = |a|$, the index of u in s. Make a sorted list of permutations $s_1, s_2, \ldots, s_{n!}$, sorted by index so that $i_{s_j} < i_{s_k}$ iff $j < k$.

Let the corresponding Hamiltonian path p visit the nodes $s_1, s_2, \ldots, s_{n!}$ in order. If e_k is the edge $s_k \to s_{k+1}$ then $w(e_k) \leq i_{k+1} - i_k$, since the factor $s[i_k, i_{k+1} + n)$ has s_k as a prefix and s_{k+1} as a suffix, hence

$$w(p) \leq \sum_{k=1}^{n-1} i_{k+1} - i_k = i_n - i_1 \leq |s| - n.$$

If this path p has an improper edge $s_k \to s_{k+1}$, the factor $s[i_k, i_{k+1} + n)$ contains some permutation other than s_k and s_{k+1} as a factor: call it v. Since we chose the last occurrence of v in s when defining i_v, we must have $i_v > i_{k+1}$. Hence the path p is well-behaved, as required. $\square$

If we can show that $w(p) \geq b$ for every well-behaved Hamiltonian path p in G_n, we may use Lemma 2 to conclude that $|s| \geq b+n$ for every superpermutation s.

2.4 The structure of the permutation digraph

In this section the variables c and d always refer to elements of N, and the variables u and v always refer to elements of N^*.

2.4.1 1-cycles

Define $\pi_1(cu) = uc$ and conversely $\rho_1(uc) = cu$.

Lemma 3. *Let $q, r \in N^*$. Then $q \sim r$ iff there exists $k \in \mathbb{Z}_{\geq 0}$ for which $\pi_1^k(q) = r$.*

Proof. If $q \sim r$ then, by definition, there exist $x, y \in N^*$ such that $q = xy$ and $r = yx$. Clearly $\pi_1^{|x|}(q) = r$. Conversely it is immediate that $z \sim \pi_1(z)$ for any $z \in N^*$, therefore

$$q \sim \pi_1(q) \sim \pi_1^2(q) \sim \cdots \sim \pi_1^k(q)$$

for every k. □

Observe that every edge of weight 1 is of the form $q \to \pi_1(q)$ – or equivalently of the form $\rho_1(q) \to q$ – for some $q \in P_n$. So every permutation is the source of one such edge, and the target of another. Thus the set of weight-1 edges constitutes a cycle cover of G_n. We shall refer to these cycles of nodes connected by weight-1 edges as **1-cycles**. Lemma 3 shows that two permutations belong to the same 1-cycle just when they are cyclically equivalent, and we shall abuse notation and write $[u]$ to denote the 1-cycle whose set of nodes is $[u]$. Each 1-cycle contains n nodes, and different 1-cycles are disjoint.

Definition 2.3. A path p is said to **cover** the 1-cycle $[x]$ if p visits every node of $[x]$.

2.4.2 2-cycles

For any c, d, u, define $\pi_2(cdu) = udc$ and conversely $\rho_2(udc) = cdu$. Every permutation u is the source of two weight-2 edges, with targets $\pi_1^2(u)$ and $\pi_2(u)$ respectively; and similarly u the target of two weight-2 edges with sources $\rho_1^2(u)$ and $\rho_2(u)$.

The most important structures in G_n are the **2-cycles**, defined as follows. Let cu be a permutation. For each such cu there is a 2-cycle, denoted $[c/u]$, whose nodes are

$$\{acb \mid a \in N^*, b \in N^*, ab \in [u]\}.$$

The edges are:

- $dv \to \pi_1(dv) = vd$, if $d \neq c$;
- $cv \to \pi_2(cv)$.

For example, if $n = 4$ then $[1/234]$ is the following cycle:

$$\begin{aligned} & 2341 \to 3412 \to 4123 \to 1234 \\ \Rightarrow\ & 3421 \to 4213 \to 2134 \to 1342 \\ \Rightarrow\ & 4231 \to 2314 \to 3142 \to 1423 \\ \Rightarrow\ & 2341 \end{aligned}$$

where $\to$ denotes an edge of the form $u \to \pi_1(u)$, and $\Rightarrow$ denotes an edge of the form $u \to \pi_2(u)$.

Definition 2.4. The **head** of the 2-cycle $[c/u]$ is c.

Remark. Some observations about 2-cycles:

- $[c/u] = [c/v]$ iff $u \sim v$;
- If $[c/u]$ and $[c/v]$ are 2-cycles, and $u \not\sim v$, then $[c/u] \cap [c/v] = \varnothing$;
- If $v \in [c/u]$, and $w \sim v$, then $w \in [c/u]$. So the set of elements of $[c/u]$ is a union of 1-cycles.

Lemma 4. *Any two 2-cycles with different heads may be written as $[c/du]$ and $[d/cv]$ where $c \neq d$.*

1. *If $u = v$ then there are two 1-cycles whose nodes belong to both 2-cycles.*
2. *If $u \neq v$ but $u \sim v$ there is one 1-cycle whose nodes belong to both 2-cycles.*
3. *If $u \not\sim v$ then the 2-cycles are disjoint.*

Proof. The 1-cycles of $[c/du]$ are $[cxdy]$, where $yx = u$, and the 1-cycles of $[d/cv]$ are $[cxdy]$ where $xy = v$.

Therefore we have a common 1-cycle of $[c/du]$ and $[d/cv]$ whenever we have words x and y such that $xy = v$ and $yx = u$.

By definition of $\sim$, this is possible only if $u \sim v$. Lemma 1 implies there is just one such pair (x, y) if $u \neq v$, and just two if $u = v$. $\square$

Definition 2.5. Given a permutation uc, the **2-cycle determined by** uc, denoted $[\![uc]\!]$, is $[c/u]$.

Definition 2.6. A path p in G_n is said to **enter the 2-cycle** t if there is an edge $e : x \to y$ in p with $w(e) > 1$ and $t = [\![y]\!]$.

Lemma 5. *If u is a permutation, then*

$$[\![pi_1(u)]\!] = [\![\pi_2(u)]\!].$$

Proof. Let $u = cdv$. Then $\pi_1(u) = dvc$, which determines the 2-cycle $[c/dv]$; and $pi_2(u) = vdc$, which determines the 2-cycle $[c/vd]$. But $[c/du] = [c/ud]$, since $du \sim ud$. $\square$

3 Lower bounds

In which we get to the point at last. The cost function is essentially the one defined by Anonymous (2011).

3.1 The cost function

Definition 3.1. Given a path

$$p = (x_0, e_1, x_1, \dots, e_{|p|}, x_{|p|})$$

in G_n, let

$$\mathrm{cost}(p) = C_0(p) + C_1(p) + C_2(p) - 2,$$

where:

- $C_0(p)$ is the number of nodes visited by p, i.e. $C_0(p) = |p| + 1$.
- $C_1(p)$ is the number of 1-cycles that are covered – in the sense of Definition 2.3 – by the prefix of p ending at $x_{|p|-1}$, i.e. by p without its final node. If $|p| = 0$ then $C_1(p) = 0$.
- $C_2(p)$ is the number of 2-cycles entered by p, in the sense of Definition 2.6.

Theorem 1. *For any path p in G_n,*

$$w(p) \geq \mathrm{cost}(p).$$

Proof. We shall use induction on the path p. If $|p| = 0$, then $C_0(p) = 1$, $C_1(p) = 0$ and $C_2(p) = 1$, hence

$$\mathrm{cost}(p) = C_0(p) + C_1(p) + C_2(p) - 2 = 0.$$

Let p be a path to the node x that has $w(p) \geq \mathrm{cost}(p)$. Let $e : x \to y$ be an edge of G_n, where p does not visit y. We need to show that

$$w(e) = w(pe) - w(p) \geq \mathrm{cost}(pe) - \mathrm{cost}(p) = (C_0(pe) - C_0(p)) + (C_1(pe) - C_1(p)) + (C_2(pe) - C_2(p)).$$

Observe:

- It is always true that $C_0(pe) - C_0(p) = 1$.
- $C_1(pe) - C_1(p)$ is 1 if p covers $[x]$, and 0 otherwise.
- $C_2(pe) - C_2(p)$ is 1 if $w(e) > 1$ and p does not enter $[\![y]\!]$, and 0 otherwise.

Now consider the possibilities for the edge e:

- If $w(e) = 1$ then $y \in [x]$, so p does not cover $[x]$ and $C_1(pe) - C_1(p) = 0$. And $C_1(pe) - C_1(p) = 0$ by Definition 2.6.
- The maximum possible value of $\mathrm{cost}(pe) - \mathrm{cost}(p)$ is 3, so if $w(e) \geq 3$ then there is nothing to prove.

This leaves the case $w(e) = 2$, which is the interesting one. It suffices to show that, if p covers $[x]$, then p must also enter $[\![y]\!]$. Suppose, then, that p covers $[x]$. There are a priori two possibilities for y given that $w(e) = 2$: either $y = \pi_2(x)$ or $y = \pi_1^2(x)$. We can rule out the latter, however, since $[\pi_1^2(x)] = [x]$ and p covers $[x]$. So $y = \pi_2(x)$.

We know that p visits $\pi_1(x)$, again since p covers $[x]$. And p necessarily visits $\pi_1(x)$ earlier than it visits x, since x is the last node of p. So p must have visited $\pi_1(x)$ from some node other than x, therefore by an edge of weight > 1. It follows that p enters $[\![\pi_1(p)]\!]$, which by Lemma 5 is equal to $[\![\pi_2(p)]\!] = [\![y]\!]$, as required. $\square$

Lemma 6. *For any Hamiltonian path p in G_n,*

$$\text{cost}(p) = n! + (n-1)! + C_2(p) - 3$$

Proof. A Hamiltonian path p must, by definition, have $C_0(p) = n!$ and $C_1(p) = (n-1)! - 1$. □

Corollary 1. *For any Hamiltonian path p in G_n,*

$$w(p) \geq n! + (n-1)! + C_2(p) - 3.$$

Remark. Since a 2-cycle contains $n(n-1)$ nodes, and a Hamiltonian path must visit $n!$ nodes, we have the trivial bound $C_2(p) \geq (n-2)!$ for a Hamiltonian path p. Therefore Corollary 1 implies immediately that

$$w(p) \geq n! + (n-1)! + (n-2)! - 3,$$

hence by Lemma 2 that

$$|s| \geq n! + (n-1)! + (n-2)! + (n-3)$$

for any superpermutation s. We shall see later that this bound is not attainable.

3.2 Slack

In order to strengthen the lower bound, we must consider the situations that increase its weight above the floor determined by the cost function. So we define:

Definition 3.2. The **slack** of a path p is the difference

$$\text{slack}(p) = w(p) - \text{cost}(p)$$

The **slack** of an edge $e : x \to y$ of a path p is

$$\text{slack}_p(e) = \text{slack}(p_y) - \text{slack}(p_x)$$

where p_x is the prefix of p ending at x, and p_y is the prefix of p ending at y.

It follows from Corollary 1 that the weight of a Hamiltonian path p depends only on:

- The number of 2-cycles entered by p, and
- How much slack p contains, i.e. the value of slack(p).

Remark. The number of 2-cycles entered by p is bounded by how those 2-cycles overlap each other. For example, the theoretical minimum $C_2(p) = (n-2)!$ is attainable only if each 2-cycle entered by p is disjoint from every other.

Remark. The known examples of low-weight Hamiltonian paths suggest there is a trade-off between the number of 2-cycles entered and the amount of slack. Paths that enter fewer 2-cycles seem to have more slack. In the next section we begin to quantify this.

4 Towards a stronger lower bound

4.1 Tight paths

Definition 4.1.

A path or cycle p is **tight** if it has no slack, i.e. if $\text{slack}(p) = 0$.

An edge e of a path p is tight if $\text{slack}_p(e) = 0$.

If p is a path and $S \subseteq \text{nodes}(p)$, say p is tight on S if $\text{slack}_p(e) = 0$ for every $e \in \text{edges}(p)$ with $\text{source}(e) \in S$.

Lemma 7. *Suppose $e : x \to y$ is a tight edge of the path peq, i.e. p is the prefix of the path ending at x, and q is the suffix of the path starting at y. Then precisely one of the following must hold:*

- $w(e) = 1$*;*
- $w(e) = 2$*, and p covers $[x]$, and p enters $[\![y]\!]$;*
- $w(e) = 2$*, and p does not cover $[x]$, and p does not enter $[\![y]\!]$;*
- $w(e) = 3$*, and p covers $[x]$, and p does not enter $[\![y]\!]$.*

Proof. This follows from Definitions 3.1, 3.2 and 4.1. TODO: more details □

Lemma 8. *Suppose $e : x \to y$ is an edge of a well-behaved Hamiltonian path peq, i.e. p is the prefix of the path ending at x, and q is the suffix of the path starting at y. The following cannot all be true:*

- *p visits $pi_1(x)$;*
- *p does not cover $[x]$;*
- *e is tight in p.*

Proof. Let us assume all these are true, and derive a contradiction. Since p visits $\pi_1(x)$, we cannot have $y = \pi_1(x)$, hence $w(e) > 1$. Since p does not cover $[x]$ and the edge e is tight, from Lemma 7 we conclude that $w(e) = 2$ and p does not enter $[\![y]\!]$.

Since p visits $\pi_1(x)$, because p is well-behaved (Definition 2.2) we cannot have $y = \pi_1^2(x)$, so we must have $y = \pi_2(x)$. But we know that p must have visited $\pi_1(x)$ by an edge of weight > 1 − since it visits x later − and therefore must have entered the 2-cycle $[\![\pi_1(x)]\!] = [\![\pi_2(x)]\!] = [\![y]\!]$. This is a contradiction. □

Lemma 9. *Let p be a well-behaved Hamiltonian path that is tight on some 1-cycle $[x]$. Every prefix of p visits a contiguous (or empty) portion of $[x]$.*

Proof. Suppose a well-behaved Hamiltonian path p has a prefix q that visits a non-contiguous portion of some 1-cycle $[x]$. We shall show that p cannot be tight on $[x]$.

Let S be the set of elements w of $[x]$ such that:

- w is not in q,
- and $\pi_1(w)$ is in q.

Since q visits a non-contiguous portion of the 1-cycle $[x]$, we have $|S| > 1$. Let w be the first node of p that belongs to S, and let r be the prefix of p ending at w. Since $S \subseteq [x]$, we have $[w] = [x]$.

We know that $S - \{w\}$ is non-empty, and that r visits no element of $S - \{w\}$, thus r does not cover $[x]$. We also know that $\pi_1(w)$ is in q, so it follows from Lemma 8 that p cannot be tight on $[x]$. □

4.2 The 2-cycle graph of a path

Definition 4.2. The **2-cycle graph** of a path p, denoted H_p, has as nodes the 2-cycles entered by p in the sense of Definition 2.6, and an edge between t and t' if t and t' share one or two 1-cycles: see Lemma 4.

Definition 4.3. Let $\mathrm{cc}(p)$ denote the number of connected components of H_p.

Lemma 10. *If p is a Hamiltonian path in G_n then*

$$C_2(p) \geq (n-2)! + (n-3)! - \frac{\mathrm{cc}(p)}{n-2}.$$

Proof. Order the nodes of H_p as

$$x_1, x_2, \ldots, x_{C_2(p)}$$

in such a way that, for every $1 < i \leq C_2(p)$, if there is any node among $x_{i+1}, \ldots, x_{C_2(p)}$ that is adjacent to some node in $x_1, \ldots, x_{i-1}$, then the node x_i is adjacent in H_p to some node in $x_1, \ldots, x_{i-1}$.

Assign each 1-cycle of G_n to the first 2-cycle in this list that contains that 1-cycle. A node that is not adjacent in H_p to any of its predecessors will be assigned all $n-1$ of the 1-cycles that make up that 2-cycle; any other node, that is adjacent to at least one of its predecessors, will be assigned at most $n-2$ 1-cycles.

The number of nodes not adjacent to any predecessor is just $\mathrm{cc}(p)$, so counting 1-cycles we have:

$$\mathrm{cc}(p) + (n-2)C_2(p) \geq (n-1)!$$

Dividing by $n-2$ gives the claimed inequality, because

$$\begin{aligned}\frac{(n-1)!}{n-2} &= (n-1)(n-3)! \\ &= ((n-2)+1)(n-3)! \\ &= (n-2)! + (n-3)!\end{aligned}$$

□

Corollary 2. *Let p be a Hamiltonian path in G_n. If the 2-cycle graph of p has no more than $k(n-2)$ connected components, for some positive integer k, then*

$$w(p) \geq n! + (n-1)! + (n-2)! + (n-3)! - (3+k) + \mathrm{slack}(p).$$

Proof. By Definition 3.2 we have

$$w(p) = \mathrm{cost}(p) + \mathrm{slack}(p),$$

and by Lemma 6 we have

$$\mathrm{cost}(p) = n! + (n-1)! + C_2(p) - 3,$$

hence

$$w(p) = n! + (n-1)! + C_2(p) - 3 + \mathrm{slack}(p),$$

and by Lemma 10,

$$w(p) \geq n! + (n-1)! + (n-2)! + (n-3)! - \frac{\mathrm{cc}(p)}{n-2} - 3 + \mathrm{slack}(p).$$

If $\mathrm{cc}(p) \leq k(n-2)$ then $\frac{\mathrm{cc}(p)}{n-2} \leq k$, and the claim follows. □

Remark. In certain cases, this bound is known to be achievable.

- Egan (2018) shows that for every n there is a Hamiltonian path p of G_n with $\mathrm{cc}(p) = 1$ and $\mathrm{slack}(p) = 0$ that has

  $$w(p) = n! + (n-1)! + (n-2)! + (n-3)! - 3,$$

  which Corollary 2 shows is the lowest possible weight for a Hamiltonian path p with $\mathrm{cc}(p) = 1$.

- For $n \leq 6$ there are known to be Hamiltonian paths p with $\mathrm{cc}(p) = n-2$, $\mathrm{slack}(p) = 0$ and

  $$w(p) = n! + (n-1)! + (n-2)! + (n-3)! - 4,$$

  which again is the lowest possible weight for a Hamiltonian path p with $\mathrm{cc}(p) = n-2$.

4.3 From 2-cycle graphs to paths

Definition 4.4. An edge of a 2-cycle graph, connecting a and b, is **simple** if a and b intersect on a single 2-cycle. A 2-cycle graph is **simple** if it is acyclic and all its edges are simple. A 2-cycle graph is **almost simple** if it fails to be simple in just one place; specifically:

- either it is acyclic and has a single non-simple edge,
- or all its edges are simple, and it has a single cycle – in the sense that there is a single edge such that removing that edge would leave an acyclic graph.

The following is really useful for showing upper bounds, rather than lower bounds, and is not proved here. But, so it is not forgotten:

Proposition 1. *Some 2-cycle graphs can always be traversed by tight paths.*

- *A simple, connected 2-cycle graph may be traversed by a tight cycle.*
- *An almost simple, connected 2-cycle graph may be traversed by a tight path.*

References

Anonymous. Lower bounds. Posted to 4chan /sci/ board. Archived at https://warosu.org/sci/thread/S3751105#p3751197, 2011.

Greg Egan. Superpermutations. http://www.gregegan.net/SCIENCE/Superpermutations/Superpermutations.html, 2018.