Tackling the Minimal Superpermutation Problem

Robin Houston

August 22, 2014

Abstract

A superpermutation on n symbols is a string that contains each of the $n!$ permutations of the n symbols as a contiguous substring. The shortest superpermutation on n symbols was conjectured to have length $\sum_{i=1}^{n} i!$. The conjecture had been verified for $n \leq 5$. We disprove it by exhibiting an explicit counterexample for $n = 6$. This counterexample was found by encoding the problem as an instance of the (asymmetric) Traveling Salesman Problem, and searching for a solution using a powerful heuristic solver.

1 The Minimal Superpermutation Problem

A superpermutation on n symbols is a string that contains each of the $n!$ permutations of the n symbols as a contiguous substring. For the sake of concreteness, we'll assume that the n symbols are the numbers $1, 2, \ldots, n$. The problem of finding minimal-length superpermutations was posed by Ashlock and Tillotson [3], who established that $\sum_{i=1}^{n} i!$ symbols suffice and conjectured that this is optimal.

There is a simple construction that takes a superpermutation on $n-1$ symbols of length k and produces a superpermutation on n symbols of length $k + n!$. Iterating this construction gives a sequence of palindromic superpermutations of length $1!, 1! + 2!, 1! + 2! + 3!, \ldots$, viz:

- `1`
- `121`
- `123121321`
- `123412314231243121342132413214321`
- `123451234152341253412354123145231425314235142315423124531243512431524312543121345213425134215342135421324513241532413524132541321453214352143251432154321`
- ...

The construction is the following. Order the $(n-1)!$ permutations of $n-1$ symbols according to the order they first appear in the superpermutation; then replace each permutation s by the n permutations of n symbols that can be obtained as cyclic shifts of sn, shifting leftwards one position at a time from

sn to ns. Finally eliminate overlap between adjacent permutations to obtain a superpermutation.

For example, the superpermutation `121` on two symbols gives the sequence `12`, `21`. The permutation `12` is expanded to `123`, `231`, `312`, and the permutation `21` expanded to `213`, `132`, `321`, so the final sequence of permutations is `123`, `231`, `312`, `213`, `132`, `321`. Overlaps are eliminated to yield `123121321`.

It is easy to show, by exhaustive enumeration using a computer, that for $n \leq 4$ these palindromic superpermutations have minimal length and are unique of this length up to relabelling. Johnston [7] showed that uniqueness fails for $n \geq 5$: there is more than one superpermutation of length $\sum_{i=1}^{n} i!$. More recently Benjamin Chaffin used a clever exhaustive computer search [1, 8] to find all eight five-symbol superpermutations of length $\sum_{i=1}^{5} i! = 153$, and show that there are no shorter ones.

The minimal length is still unknown for $n \geq 6$, but we can show that for all $n \geq 6$ it is strictly less than the conjectured length $\sum_{i=1}^{n} i!$.

2 The Travelling Salesman Problem

The Travelling Salesman Problem (TSP) is the problem of finding a minimum-weight Hamiltonian circuit of a weighted graph. We are interested mainly in the asymmetric TSP, where the graph is directed.

For an n-symbol alphabet, we may construct a complete directed graph with $n!$ vertices, one for each permutation of the n symbols. The weight of the edge from s to t is the least $0 \leq k \leq n$ such that the $(n-k)$-suffix of s is equal to the $(n-k)$-prefix of t. Then a minimum-weight Hamiltonian path in this graph corresponds to a minimal superpermutation on n symbols. The length of the corresponding superpermutation is $n + w$ where w is the weight of the minimum-weight Hamiltonian path.

So far this is not quite an instance of the TSP, since we are looking for a Hamiltonian *path* rather than a Hamiltonian circuit. To relate it to the TSP it is sufficient to change the weights of certain edges to 0. Specifically, let o be the identity permutation $12 \ldots n$ and let the weight of each edge s–o be 0. A minimum-weight Hamiltonian circuit on the resulting graph corresponds to a minimum-weight Hamiltonian path starting at o.

The TSP has been intensively studied, and though it is NP-hard − hence no worst-case polynomial-time algorithm is known − there are solvers that in practice work very well. In particular we used Concorde [2], which finds provably-minimal solutions; and LKH [5, 6], which is a fast, randomised, approximate solver.

Concorde only works for the symmetric TSP, so to apply Concorde it was necessary to use the Jonker-Volgenant construction [9, 10] to convert our instance of the asymmetric TSP on $n!$ vertices to an instance of the symmetric TSP on $2(n!)$ vertices.

We were able to use Concorde to solve the five-symbol instance in 1929.02 seconds on an Amazon EC2 'm3.medium' instance running Linux, confirming Chaffin's recent result. We have not found an exact solution to the six-symbol instance − Concorde failed with an internal error after running for several days. Nevertheless we conjecture it is within reach of current technology.

On the other hand, we were able to falsify the minimal-length conjecture on six symbols without solving the instance completely, using LKH to search repeatedly for approximate solutions till we found a Hamiltonian circuit of weight 866, representing a superpermutation of length 872 (less than the conjectured minimum of 873). Since LKH supports the asymmetric TSP directly, it was not necessary to use the larger symmetrical instance in this case. The first superpermutation of length 872 was found on the 9228th run – with each run taking approximately 2–4 seconds on a 2010 MacBook Pro with 2.66 GHz Intel Core 2 Duo processor. The parameters used were:

```
BACKTRACKING = YES
MAX_CANDIDATES = 6 SYMMETRIC
MOVE_TYPE = 3
PATCHING_C = 3
PATCHING_A = 2
```

These parameters are those used by Helsgaun to find optimal solutions to David Soler's ATSP instances [4], with the addition of the `BACKTRACKING` option.

3 A short superpermutation on six symbols

The following superpermutation on six symbols has length 872, which is less than the conjectured minimum of $\sum_{i=1}^{6} i! = 873$. The recursive construction described in Section 1 may then be used to construct a superpermutation shorter than the conjectured minimum for all $n > 6$.

```
123456123451623451263451236451326451362451364251364521364512346512341 5
```

4 Remarks

General-purpose solvers for NP-hard problems – SAT, SMT, TSP, etc. – are remarkably powerful and applicable to many combinatorial problems. The world of pure mathematics has been slow to exploit this opportunity, but that is beginning to change: the recent advances on the Erdős Discrepancy Problem using a SAT solver [11, 12] are especially notable in this regard.

It seems plausible that the minimal superpermutation problem for $n = 6$ could be solved exactly using Concorde or a similar algorithm with a few weeks or months of CPU time. More generally, these instances seem to present an interesting challenge for the TSP community.

5 Ancillary files

Several ancillary files are included, listed with the arXiv abstract.

Python scripts and their output

- `mkatsp.py` Generate an instance of the asymmetric TSP representing the minimal superpermutation problem for a given value of n. The output is a TSPLIB file of type ATSP. The files `5.atsp` and `6.atsp` are the output of this script for $n = 5$ and $n = 6$.
- `symmetrise.py` Apply the Jonker-Volgenant transformation to transform an ATSP instance into a corresponding symmetric TSP instance. The files `5.tsp` and `6.tsp` are the result of applying this to `5.atsp` and `6.atsp` respectively.
- `mkpalindromic.py` Produce a simple palindromic superpermutation on the specified number of symbols.
- `printtour.py` Take a tour produced by LKH and print the corresponding superpermutation.
- `splitsuperperm.py` Split a superpermutation into permutations. This is useful for analysing the structure of a superpermutation, and (by sorting the output) for verifying that a supposed superpermutation really does produce all permutations.

Other files

- `concorde-output-5-symbols.txt` The log of running Concorde on `5.tsp`.
- `6.par` The parameter file used to run LKH on `5.atsp`.
- `6.866.lkh` The first tour of weight 866 obtained when running LKH with parameter file `5.par`.
- `superperm-6-866.txt` The superpermutation of length 872, obtained by running `printtour.py` on `6.866.lkh`.

References

[1] Sequence A180632 in the On-Line Encyclopedia of Integer Sequences. `http://oeis.org/A180632`.

[2] David Applegate, Ribert Bixby, Vasek Chvatal, and William Cook. Concorde TSP solver. `http://www.math.uwaterloo.ca/tsp/concorde.html`, 2006.

[3] Daniel A. Ashlock and Jenett Tillotson. Construction of small superpermutations and minimal injective superstrings. *Congressus Numerantium*, 93:91–98, 1993.

[4] Keld Helsgaun. LKH results for Soler's ATSP instances. `http://www.akira.ruc.dk/~keld/research/LKH/Soler_results.html`.

[5] Keld Helsgaun. An effective implementation of the Lin-Kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126:106–130, 2000.

[6] Keld Helsgaun. General k-opt submoves for the Lin-Kernighan TSP heuristic. *Mathematical Programming Computation*, 1(2-3):119–163, 2009.

[7] Nathaniel Johnston. Non-uniqueness of minimal superpermutations. *Discrete Mathematics*, 313(14):1553–1557, 2013. arXiv:1303.4150.

[8] Nathaniel Johnston. All minimal superpermutations on five symbols have been found. `http://www.njohnston.ca/2014/08/all-minimal-superpermutations-on-five-symbols-have-been-found/`, 2014.

[9] Roy Jonker and Ton Volgenant. Transforming asymmetric into symmetric traveling salesman problems. *Operations Research Letters*, 2(4):161–163, 1983.

[10] Roy Jonker and Ton Volgenant. Transforming asymmetric into symmetric traveling salesman problems: erratum. *Operations Research Letters*, 5(4):215–216, 1986.

[11] Boris Konev and Alexei Lisitsa. Computer-aided proof of Erdős discrepancy properties. `http://cgi.csc.liv.ac.uk/~konev/edp/`.

[12] Boris Konev and Alexei Lisitsa. A SAT attack on the Erdős discrepancy conjecture. `http://arXiv.org/abs/1402.2184`, 2014.