

2004

Pokémon® Cards and the Shortest Common Superstring

Mark Stamp

San Jose State University, mark.stamp@sjsu.edu

Austin Stamp

Follow this and additional works at: https://scholarworks.sjsu.edu/computer_sci_pub



Part of the [Computer Sciences Commons](#), and the [Other Mathematics Commons](#)

Recommended Citation

Mark Stamp and Austin Stamp. "Pokémon® Cards and the Shortest Common Superstring" *Graph Theory Notes of New York* (2004): 19-24.

This Article is brought to you for free and open access by the Computer Science at SJSU ScholarWorks. It has been accepted for inclusion in Faculty Publications, Computer Science by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

[GTN XLVII:3]

POKÉMON CARDS AND THE SHORTEST COMMON SUPERSTRING

Mark Stamp and Austin E Stamp

5827 Tompkins Drive
San Jose, California 95129, U.S.A.
<stamp@cs.sjsu.edu>

Abstract

Evidence is presented that certain sequences of Pokémon cards are determined by selecting consecutive elements from a longer sequence. We then consider the problem of recovering the shortest common superstring (SCS), that is, the shortest string that contains each of the Pokémon card sequences as a consecutive substring. The SCS problem arises in many applications, most notably in DNA sequencing.

1. Introduction

Pokémon are mythical bug-like creatures invented by Satoshi Tajiri [1]. They were originally known as *poketto monstaa*, Japanese for *pocket monster*, a name that was eventually shortened to Pokémon.

Pokémon first appeared as characters in a Game Boy game in 1996, a game that was available only in Japan. This, was followed by a Japanese cartoon, or anime, series (see [2] for an unusual story concerning the original Pokémon anime series). The game and television series, became world-wide successes.

In 1999 a highly successful Pokémon trading card game was introduced. This game is sometimes described as a sophisticated form of the *rock, paper, and scissors* game. In the Pokémon version, each trainer (that is, player) has a set of trading cards that are used to battle the cards of an opposing trainer. So-called booster packs of cards are available so that a trainer can create decks of cards with various characteristics.

2. Booster Packs

We consider 11-card Pokémon booster packs from the *base one, unlimited edition series*. In this series there are 102 distinct cards, consisting of 32 common cards, 6 energy cards, 32 uncommon cards, 16 rare cards, and 16 rare holofoil cards. The 32 common cards are numbered 43 through 69 (Pokémon cards) and 91 through 95 (trainer cards), the energy cards are numbered 97 through 102, the uncommon cards are 23 through 42 (Pokémon) and 80 through 90 (trainer) and 96 (uncommon energy), the rare holofoil cards are numbered 1 through 16, and the rare non-holofoil cards are 17 through 22 (Pokémon) and 70 through 79 (trainer).

Each 11-card booster pack contains five common cards, two energy cards, one rare card, and three uncommon cards. Each type of card occurs consecutively as a block, but the order in which these blocks appear can vary from one printing to another. See [3] for more details about Pokémon and the trading cards considered in this paper.

3. Shortest Common Superstring

Consider the following four sequences, or strings, of common Pokémon cards:

$A = (95, 64, 44, 53, 57)$

$B = (64, 44, 53, 57, 59)$

$C = (44, 53, 57, 59, 91)$

$D = (91, 56, 46, 45, 52).$

Observe that strings A and B overlap and the *superstring* $(95, 64, 44, 53, 57, 59)$ is the shortest string that contains both A and B .

For common Pokémon cards, we consider the problem of finding the *shortest common superstring*, that is, the superstring of shortest length that contains all of the common card sequences as consecutive substrings. The

shortest common superstring problem arises in many applications, including *DNA sequencing* and *data compression* [4].

Let $o(X, Y)$ be the maximum overlap when string X is followed by string Y . To solve the shortest common superstring problem for the four common Pokémon card strings listed above we first construct, the complete directed graph G , shown in Figure 1, in which the arc from node X to node Y is given the edge weight $o(X, Y)$. Then a solution to the shortest common superstring problem is a *directed Hamilton path* (not necessarily a circuit) in G of maximum total arc weight.

For the graph shown in Figure 1 the optimal solution is

$$A \rightarrow B \rightarrow C \rightarrow D,$$

which has total arc weight 9 and consequently yields an shortest common superstring of length 11. The shortest common superstring is given as follows, with the initial point of each of the four strings as indicated:

$$(1) \quad \begin{array}{cccc} 95, 64, 44, 53, 57, 59, 91, 56, 46, 45, 52 \\ A & B & C & D \end{array}$$

As an aside, it is interesting to note that the *traveling salesman problem* can be formulated in a manner very similar to the shortest common superstring problem. The significant differences between the two problems are that for the traveling salesman problem the underlying graph is undirected and the goal is to find the Hamilton path of minimum total edge weight [5].

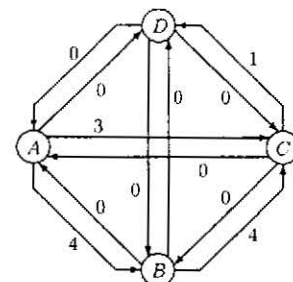


Figure 1: Directed graph G .

4. Common Card Strings

We examined 153 booster packs, carefully recording the common card string that appears in each pack. The resulting 153 common card strings contain a large number of duplicates, leaving only 88 distinct strings. Three of these 88 appear to have a pair of elements transposed. We believe this to be the case since the offending pairs do not appear within any other strings. However, if swapped, each appears as a substring multiple times within other strings. Thus, we consider a set of 85 distinct strings, each of length five.

Recall that there are 32 distinct common Pokémon cards. In the 85 strings under consideration each of the 32 distinct cards is succeeded by at most four different cards. This observation led us to conjecture that the common cards are drawn from a superstring of length 128, with each of the 32 distinct cards appearing four times within the superstring. Here we provide evidence that this is (approximately) the case.

Let N denote the number of strings under consideration, L be the length of the underlying superstring, n be the number of distinct symbols, and let m be the length of each string. We assume that all strings are of the same length. In the Pokémon problem we have $N = 85$, $n = 32$, $m = 5$, and L is to be determined. We assume that the strings are not allowed to "wrap", and hence, there are $L - m + 1$ positions at which the initial point of a string can occur.

Since $N < L - m + 1$, we can interpret

$$(2) \quad p = \frac{N}{L - m + 1}$$

as the probability of a sequence starting at a given point within the superstring and $1 - p$ as the probability that no string begins at a specified position. It follows that the expected number of strings that overlap in k positions is about

$$(3) \quad \frac{N^2}{L - m + 1}$$

and this holds for any $k \in \{1, 2, \dots, m - 1\}$. Since we do not allow duplicate sequences $k = m$ cannot occur.

Let s_i be the i^{th} Pokémon card string. Define $\mathcal{O}$ to be the 85×85 overlap matrix given by $\mathcal{O}_{ij} = o(s_i, s_j)$. For the 85 sequences under consideration, the values in $\mathcal{O}$ are summarized in the following Table, where we have ignored the diagonal elements.

Table: Summary of overlap matrix $\mathcal{O}$

Overlap	Number
0	6744
1	225
2	54
3	56
4	61

Suppose that the Pokémon common card strings are selected at random (with replacement, for simplicity) and each symbol is equiprobable. Then the expected number of strings with overlap of length two is approximately

$$\frac{N(N-1)}{n^2},$$

which, for $N = 85$ and $n = 32$, is about 7. A similar calculation shows that the expected number of strings with overlap of length three or four are both less than one.

We therefore expect, at random, to find about 7 pairs with an overlap of length two and no such random matches for any length greater than two. In the Table we see that the number of pairs with overlap of length two is actually less than the number with overlap three or four. One explanation for this is that every overlap of length 3 or 4 is causal (i.e., drawn from the superstring) and is not a random match. This could occur if the superstring was chosen so that no consecutive pair of elements is repeated within the superstring. We briefly return to this point in the next section.

Ignoring any random overlap of length two, we average the number of overlaps, of length 2, 3, and 4 from the Table and obtain the value 57. We then set, Expression (3) equal to 57 and solve for the unknown value L , to find $L = 130.75$. This result is close to our conjectured value, $L = 128$. These calculations indicate that the Pokémon card strings are not randomly selected. Furthermore, the numbers are consistent with our conjecture that the strings are selected from a superstring of about 128 elements.

As mentioned already, any overlap of length three or four is, with high probability, casual. However, an overlap of length three, for example, might not contribute any new information. In particular, if an overlap of length three coincides with two pairs of strings that have overlap of length four, then this only serves to reinforce information that is already known to be almost certainly correct. For example, consider the three strings A , B , and C from Figure 1. We have $o(A, B) = o(B, C) = 4$ and $o(A, C) = 3$. However, the overlap of length three does not contribute any useful information. Of course, a similar situation can arise with strings that overlap in fewer than three elements. This motivates the following definition.

Definition 1: We refer to pairs of strings that have the maximum overlap within a block as *relevant*; those that do not are *irrelevant*. ■

In Figure 1, the pairs (A, B) and (B, C) are relevant, whereas (A, C) is irrelevant.

The expected number of relevant pairs of strings can be estimated. For example, a pair of strings with overlap of length three is relevant provided that no string begins at the one superstring position that lies between the starting points of the pair of strings. The probability of this is $1 - p$, where p is given in (2).

In the Pokémon problem all pairs of strings with overlap of length three are almost certainly causal, but only about $56(1 - 85/124)$ or about 18 of these will be relevant. Similarly, about six of the overlaps of length two and about two of the overlaps of length one are relevant. In addition, we expect less than one of the overlaps of length zero to be relevant. To have any hope of recovering the true shortest common superstring we must have no relevant overlap of length zero, otherwise we would need to select matching pairs of strings where no score is available.

In general, about

$$(4) \quad \frac{N^2}{L-m+1} \left(1 - \frac{N}{L-m+1}\right)^{m-k-1}$$

of the causal k -overlapping pairs will be relevant. To simplify the recovery of the Pokémon shortest common superstring it would be useful to have no relevant overlaps of length one. In light of (4), and assuming $L = 128$, we could attempt to increase N to the point where

$$\frac{N^2}{124} \left(1 - \frac{N}{124}\right)^3 < 1.$$

We find that any $N \geq 95$ will suffice. Similarly, we would need $N \geq 112$ before the expected number of relevant overlaps of length two would fall below one. With the 85 available Pokémon strings we expect to be able to construct a shortest common superstring for the Pokémon problem using only overlaps of length one and greater. In the next section we make use of this observation to devise a simple *greedy algorithm* to solve the Pokémon shortest common superstring problem.

5. A Greedy Solution

Suppose we have N strings, s_0 through s_{N-1} . Then one approach to finding the exact shortest common superstring solution would be to compute the overlap score associated with each of the $N!$ permutations of $\{s_0, s_1, s_2, \dots, s_{N-1}\}$. A permutation with maximum score would yield a solution to the shortest common superstring problem. Of course, the work factor for this approach is prohibitive, even for relatively small values of N .

As a decision problem, the shortest common superstring, is known to be NP-complete [6]. Hence, in general, we cannot expect to find the exact shortest common superstring solution efficiently. However, there are efficient techniques that are known to give good approximate solutions. For example, dynamic programming or a greedy algorithm will produce a solution whose length is within a small multiple of the shortest common superstring [4][7][8].

For the Pokémon shortest common superstring problem the 85×85 overlap matrix $\mathcal{O}$ is sparse, containing only about 400 nonzero entries. Furthermore, as the Table indicates, all pairs with overlap two or greater are likely causal. As discussed in the previous section, we also expect all but about two of the relevant overlaps to be of length two or greater.

Given these observations, we implemented a simple greedy approach that includes branching in the case of all tie scores meeting or exceeding a given threshold. The algorithm begins by initiating a permutation for each of the N strings,

$$\text{perm}[i][0] = i, \text{ for } i = 0, 1, \dots, N-1.$$

Then we proceed to greedily construct a complete permutation out of each partial permutation by appending the eligible string with maximum overlap at each step. However, if there is more than one candidate string that attains the maximum overlap at a given step and that overlap meets or exceeds, a bound k , we create a new partial permutation (or branch) for each such string.

From the definition of relevant strings it is clear that this greedy algorithm will yield all possible shortest common superstrings provided k is chosen so that all relevant overlaps are of length k or greater. We state this result as a theorem.

Theorem 1: A greedy algorithm with branching on all ties of overlap k or greater will yield a shortest common superstring for the given data set provided there is no relevant overlap of length less than k . ■

The work factor for this greedy branching algorithm depends on the number of ties that meet or exceed the bound k . Generating and scoring any one permutation of length N requires, on the order of N^2 work. The total work is therefore bounded by MN^2 , where M is the final number of permutations (i.e., superstrings). If we have chosen k correctly, then the highest scoring of these M superstrings will necessarily yield a shortest common superstring. We can obtain an estimate for k by using the methods of the previous section.

Initially there are N partial permutations. With p as given in (2), the probability that any (partial) permutation has a relevant overlap of length k is

Each Pokémon common card number occurs four times in the superstring (5), except for 67, 69, 91, 92, 94, and 95, each of which appears three times. Consequently, the length of the superstring is slightly less than conjectured. It is possible that this is simply due to a lack of data at either or both ends of the superstring. It is also the case that every overlap of length two or greater is causal. Apparently, the superstring was carefully chosen so as to avoid any repeated two-element patterns. In addition, any five-element string selected from the superstring (5) does not contain duplicate card numbers. This feature insures distinct common cards within a Pokémon card booster pack.

6. Data

The complete set of data for the Pokémon cards analyzed in this paper is available at

<http://home.earthlink.net/~mstamp1/pokemon/pokemondata.html>.

More data will be added as time and budget permits.

References

- [1] Pokémon: A quick history, at <http://members.tripod.com/animefan25/pkmnhistor.html>.
- [2] The Pokémon seizure incident. at <http://members.tripod.com/animefan25/seizures.html>.
- [3] M. Stamp and A. Stamp; Pokémon trading card sequences, at <http://home.earthlink.net/~mstamp1/papers/pokemon.pdf>.
- [4] A. Frieze and V. Szpankowski; Greedy algorithms for the shortest common superstring that are asymptotically optimal. *Algorithmica*, **21**, 21–36 (1998).
- [5] D.L. Kreher and D.R. Stinson; *Combinatorial Algorithms: Generation, Enumeration, and Search*. CRC Press, Boca Raton (1999).
- [6] M.R. Garey and D.S. Johnson; *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 228 (1979).
- [7] A.G. Percus and D.C. Torney; Greedy algorithms for optimized DNA sequencing, Proceedings of the Tenth Annual ACM–SIAM Symposium on Discrete Algorithms (SODA 1999), 955–956 (1999).
- [8] J.S. Turner; Approximation algorithms for the shortest common superstring problem. *Information and Computation*, **83**(1), 21–40 (1989).

Received: June 24, 2003