Isomorphism Testing of Steiner Triple Systems: Canonical Forms

D.R. Stinson

ABSTRACT

A method for testing isomorphism of Steiner triple systems in time $0(v^{\log v})$ is improved and implemented. We conjecture that the method works in time $0(v^4 \log v)$ on average. Empirical evidence supports this conjecture.

1. Introduction.

A *Steiner triple system* of *order* v is a set $\mathbf{B}$ of $b = v(v-1)/6$ unordered 3-subsets *(blocks)*, chosen from a set (of *points)* $\{1,2,...,v\}$ in such a way that every unordered pair of points occurs in a unique block. We will abbreviate the term Steiner triple system to STS. It can easily be shown that an STS of order v exists only if $v \equiv 1$ or 3 mod 6. This necessary condition for existence was shown to be sufficient by Kirkman [4] in 1847.

Suppose $\mathbf{B}_1$ and $\mathbf{B}_2$ are two STS of order v. We say that $\mathbf{B}_1$ and $\mathbf{B}_2$ are *isomorphic* if there exists a permutation Π of $\{1,2,...,v\}$ such that $\{x,y,z\} \in \mathbf{B}_1$ if and only if $\{\Pi(x),\Pi(y),\Pi(z)\} \in \mathbf{B}_2$. Isomorphism testing is usually done by means of invariants. An *invariant* is a mapping f, defined on the set of all STS, such that $f(\mathbf{B}_1) = f(\mathbf{B}_2)$ if $\mathbf{B}_1$ and $\mathbf{B}_2$ are isomorphic. The image $f(\mathbf{B})$ of an STS $\mathbf{B}$ is called the *form* of $\mathbf{B}$. The use of invariants is most useful when testing several STS for isomorphism. If we want to test d STS, we can calculate the d forms, and then sort them (in time $0(d \log d)$). This provides a significant time saving over testing all $\binom{d}{2}$ pairs of STS directly for isomorphism.

The difficulty with most invariants is that two STS may have the same forms, yet not be non-isomorphic. See, for example, [1] and [3].

The method we investigate in this paper is a *complete* invariant: two STS are isomorphic if and only if they have the same forms. We refer to this invariant as the *canonical form*. In section 2, we describe the basic algorithm, due to Miller [8]. In section 3, we discuss methods of improving

ARS COMBINATORIA, Vol. 19 (1985), pp. 213-218

the performance of the algorithm, and examine its behaviour in practice. In section 4, we compare the method of canonical forms to similar techniques used in graph theory.

2. Canonical forms and labellings.

The basic method described in this section is due to Miller [8]; in [1], Colbourn describes a generalization to Steiner systems $S(t,t+1,v)$.

Let $B_1 = \{x_1,y_1,z_1\}$ and $B_2 = \{x_2,y_2,z_2\}$ be two 3-subsets of $\{1,2,\ldots,v\}$, where $x_1 < y_1 < z_1$ and $x_2 < y_2 < z_2$. We say that $B_1 < B_2$ if $z_1 < z_2$; or $z_1 = z_2$ and $y_1 < y_2$; or $z_1 = z_2$, $y_1 = y_2$, and $x_1 < x_2$. An *ordered* STS is a list $\mathbf{B} = (B_1,\ldots,B_b)$ of blocks such that $\{B_1,\ldots,B_b\}$ is an STS, and $B_1 < B_2 < \cdots < B_b$. Let $\mathbf{B}$ and $\mathbf{B}'$ be two ordered STS (of order v), where $\mathbf{B} = (B_i: 1 \le i \le b)$ and $\mathbf{B}' = (B'_i: 1 \le i \le b)$. We say that $\mathbf{B} < \mathbf{B}'$ if there is a j $(1 \le j \le b)$ such that $B_i = B'_i$ for $1 \le i < j$, and $B_j < B'_j$.

Given an STS of order v, $\mathbf{B}$, one can produce $v!$ isomorphic STS from it by permuting the v points in all possible ways. Of the $v!$ corresponding ordered STS, the least is denoted $f(\mathbf{B})$ and called the *canonical form* of $\mathbf{B}$. The permutation Π which gives rise to it is called the *canonical labelling*. The following result can be easily proved.

Theorem 2.1. *A canonical form is a complete invariant for Steiner triple systems.*

Calculation of $f(\mathbf{B})$ by naive methods requires exponential time. However, we do not need to consider all $v!$ permutations of points to find the canonical form. Define a *partial labelling* to be a partial permutation Π of $\{1,\ldots,v\}$ where $\{\Pi(i): i \in dom\ (\Pi)\} = \{i: 1 \le i \le |dom\ (\Pi)|\}$. A point i in $dom\ (\Pi)$ is said to be *labelled;* its *label* is $\Pi(i)$.

For any block $B = \{x,y,z\}$ in $\mathbf{B}$ with precisely two labelled points, x and y, define $\Pi(B) = \{\Pi(x),\Pi(y),v\}$. Now suppose that Π_1 is a canonical labelling which extends Π, and suppose there is at least one block with precisely two labelled points. Choose this block B so that $\Pi(B)$ is minimized. Then $\Pi_1(z) = |dom\ (\Pi)| + 1$, since the forms arising from permutations Π_1 extending Π will first differ in the way B is labelled. The number $|dom\ (\Pi)| + 1$ is the smallest available label. We can now describe Miller's algorithm.

```
$\Pi$: = null partial permutation;
while $|dom\ (\Pi)| < v$ do
    begin
        choose any unlabelled point $x$;
        $\Pi(x)$:= $|dom\ (\Pi)| + 1$;
    while there is a block with precisely two labelled points do
        begin
            of those blocks, let $B$ be that one which minimizes $\Pi(B)$;
            if $z$ is the unlabelled point in $B$, then $\Pi(z) := |\, dom\ (\Pi) | + 1$
        end
end
```

A form produced by the above algorithm is said to be *legitimate.* By the preceding discussion, we have

Theorem 2.2. *The canonical form of an STS is legitimate.*

Thus, in order to calculate the canonical form, we need only calculate all legitimate forms, and find the least of them. It is not hard to see that any legitimate form can be produced in polynomial time (we will be more precise later). The immediate question is: how many legitimate labellings does an STS have? This depends on the number of times we are faced to choose an unlabelled point x.

The only time we must choose a point is when $|\, dom\ (\Pi) \cap B | \neq 2$ for any block B. Thus $\mathbf{B}_\Pi = \{B: B \subseteq dom\ (\Pi)\}$ forms a STS in its own right on pont set *dom* (Π). (We say that $\mathbf{B}_\Pi$ is a *sub-STS).* In the process of producing a legitimate labelling we build up a sequence of sub-STS, each contained in the next. It is not difficult to prove that is an STS of order v contains a sub-STS of order w, then $v \geq 2w + 1$. Hence, there can be no more than log v "nested" sub-STS encountered in producing a legitimate labelling. At any time, there cannot be any more than v choices for the next labelled point. Thus we obtain

Theorem 2.3. *There are* $0(v^{\log v})$ *legitimate forms.*

In fact, the projective space $PG(n,2)$, which is an STS of order $v = 2^{n+1} - 1$, has $\theta(v^{\log v})$ legitimate forms. However, one can "recognize" $PG(n,2)$ in time $0(v^3)$, by checking Pasch's axiom (given two intersecting blocks $\{x,y_1,z_1\}$ and $\{x,y_2,z_2\}$, the two blocks containing y_1 and y_2, and z_1 and z_2, should intersect in a point).

Theorem 2.3 provides a worst-case bound, but the average-case appears to be much better. An STS with no sub-STS, other than single blocks, is said to be *planar.* It is easy to prove

Theorem 2.4. *A planar STS of order v has $n(n-1)(n-3)$ legitimate forms.*

There is a conjecture, due to Quackenbush [9], that almost all triple systems are planar. If this conjecture is true, it would be likely that one could test isomorphism of STS, on average, in polynomial time.

3. An improvement.

We can improve on the basic method, as follows. Suppose we have an STS, say **B**, and a partial labelling Π. A block B is said to be *labelled* if $B \subseteq dom\ (\Pi)$. The idea is to implement the algorithm for generating a legitimate form so that blocks become labelled *in order*. That is, if $B_1 = \{x_1,y_1,z_1\}$ is labelled before $B_2 = \{x_2,y_2,z_2\}$, then $\{\Pi(x_1),\Pi(y_1),\Pi(z_1)\} < \{\Pi(x_2),\Pi(y_2),\Pi(z_2)\}$.

If legitimate forms are generated in this manner, then it can happen (and it usually does) that we can determine that a given partial labelling can not be extended to any canonical labelling. We simply keep track of the "best" form at any stage of the algorithm. As we generate a new legitimate form, we compare each labelled block of the new form to the corresponding block of the "best" form. If this new block is greater, we can quit, for we will not improve on the "best" form, and try building another legitimate form. If the new block is less than the one in the "best" form, we know that the new form we are constructing will improve upon the previous "best" form, and so it will become the "best" form when we have finished constructing it.

In the case of planar STS, it is not difficult to find the canonical form in time $0(v^5)$. Indeed, if we actually construct all legitimate forms, the process requires time $\theta(v^5)$. The above approach enables us to determine the canonical form without actually constructing all legitimate forms. We suspect, but cannot prove, that this will reduce the running time to $0(v^4\log v)$. In any event, we can do no better than $\Omega(v^4)$ by this approach.

We programmed this algorithm in PASCAL/VS, and ran it on the University of Manitoba AMDAHL 5850 computer. STS were generated by means of a hill-climbing algorithm described in [11]. This hill-climbing algorithm enables one to generate many STS in a very short time. It appears to work in time $0(v^2\log v)$, although we know of no proof that the algorithm will succeed in any amount of time! We hope that STS generated by this approach are "random" in some sense. (See [11], for a discussion.)

The following timings were obtained.

Table 1

Order	average time to find canonical form (sec.)
15	1.2
19	3.4
21	5.0
25	8.5
27	11.1
31	18.4

The above timings are consistent with our conjecture of $0(v^4 \log v)$, and indeed, are consistent with $0(v^4)$.

For purposes of comparison, we also generated canonical forms of STS by means of the general-purpose graph isomorphism programs of B. McKay [6] and W.L. Kocay [5]. As far as the author is aware, these are the fastest general-purpose programs in existence. One deals with STS in this setting by constructing a bipartite graph (with vertices being the blocks and points of the STS), joining a block and a point if and only if the point is a member of the block. Such a graph is equivalent to the STS.

These programs were run on STS of order 15, on the same machine. The time taken for Kocay's program was an average of 6.5 sec., and for McKay's program 17.8 sec. Thus our approach is significantly faster. This is, of course, due to the fact that we are investigating a more specialized problem, and we were able to find improvements that could not apply in the general situation of graph testing. However, we should note that the testing of balanced incomplete block designs has traditionally been regarded as a difficult subcase of graph testing (see Mathon [7]).

4. Remarks.

The graph isomorphism problem has received considerable attention (see, for example, [5], [6] and [10]). No sub-exponential algorithm is known, nor is the problem known to be NP-complete. The special case of testing isomorphism of balanced incomplete block designs (BIBDs) is isomorphism complete: it is as difficult as the general problem (this was proved by Colbourn and Colbourn [2]). An STS is just a BIBD with block-size 3 and $\lambda = 1$. As we have seen, there is an $0(v^{\log v})$ algorithm for testing isomorphism of STS, which is subexponential. We have suggested that the average case behaviour of this algorithm may in fact be $0(v^4 \log v)$.

References.

[1] M.J. Colbourn, *An analysis technique for Steiner triple systems*, Proc. Tenth S.E. Conf. on Combinatorics, Graph theory and computing (1979), 289-303.

[2] M.J. Colbourn and C.J. Colbourn, *Concerning the complexity of deciding isomorphism of block designs*, Discrete Applied Math. 3 (1981), 155-162.

[3] M.J. Colbourn, C.J. Colbourn and W.L. Rosenbaum, *Trains: an invariant for Steiner triple systems*, Ars Combinatoria 13 (1982), 149-162.

[4] T.P. Kirkman, *On a problem in combinations*, Cambridge and Dublin Math. Journal 2 (1847), 191-204.

[5] W.L. Kocay, *Abstract data types and graph isomorphism*, Journal of Comb., Inf. and Sys. Sci., to appear.

[6] B.D. McKay, *Practical graph isomorphism*, Congressus Numerantium 30 (1981), 45-87.

[7] R. Mathon, *Sample graphs for isomorphism testing*, Proc. Ninth S.E. Conf. on Combinatorics, Graph theory and Computing (1978), 499-517.

[8] G.L. Miller, *On the $n^{\log n}$ isomorphism technique*, Proc. Tenth Annual A.C.M. Symposium on the Theory of Computing (1978), 51-58.

[9] R.W. Quackenbush, *Algebraic speculations about Steiner systems*, Annals of Discrete Math. 7 (1980), 25-35.

[10] R.C. Read and D.G. Corneil, *The graph isomorphism disease*, J. Graph Theory 1 (1977), 339-363.

[11] D.R. Stinson, *Hill-climbing algorithms for the construction of combinatorial designs*, Annals of Discrete Math., to appear.

Department of Computer Science
University of Manitoba
Winnipeg, Manitoba
R3T 2N2 Canada