On the $n^{\log n}$ Isomorphism Technique*†

A preliminary report

Gary L. Miller
Department of Mathematics
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Tarjan has given an algorithm for deciding isomorphism of two groups of order n (given as multiplication tables) which runs in $O(n^{(\log_2 n + O(1)}$ steps where n is the order of the groups. Tarjan uses the fact that a group of n is generated by log n elements. In this paper, we show that Tarjan's technique generalizes to isomorphism of quasigroups, latin squares, Steiner systems, and many graphs generated from these combinatorial objects.

Introduction

One of the original papers on algorithms for graph isomorphism is by Corneil and Gotlieb [4]. In this paper they indicate that a possible computationally difficult subproblem of the graph isomorphism problem is when the graphs are strongly regular. Since this paper, Corneil and Mathon have gathered together a collection of strongly regular graphs which many researchers [4,8] have used as canonically difficult graphs for isomorphism testing.

A graph is said to be strongly regular SR if there exist three constants κ, λ, μ such that (1) κ is the valence of each vertex, (2) λ is the number of vertices common to each pair of adjacent vertices, and (3) μ is the number of vertices common to each pair of nonadjacent vertices. The parameter set for a strongly regular graph is often denoted by (n,κ,λ,μ) where n is the number of vertices.

One important method for generating SR graphs is to take a combinatorial structure and construct a graph associated with it. Essentially, in this paper we shall consider two combinatorial structures, latin squares and Steiner triple systems, and their associated SR graphs called latin square graphs and Steiner triple system graphs. These graphs will have a certain parameter set. Any SR graph which has the same parameter set as a latin square graph will be called a pseudo-latin square graph; similarly for Steiner triple systems. In fact, these pseudo-latin square graphs and pseudo-Steiner graphs are the particular graphs used to experimentally analyze the possible worst case behavior of isomorphism algorithms.

In this paper, we shall show in particular that latin square and Steiner triple system graphs can be decided in $O(n^{\log_2 n + O(1)})$ steps and show, theoretically at least, that pseudo-latin square and pseudo-Steiner triple system graphs are also decidable in this time, using some results of Bruck and others [1,2,3].

In Section 1 we introduce the $n^{\log n}$ technique, and show how to apply it to generalizations of groups and finally to latin square graphs. In Section 2 we discuss the implications of Bruck's

*This research was in part supported by the National Research Council, Grant #NRC-A5549; in part by the Alfred P. Sloan Foundation under Grant 74-12-5; and in part by the Advanced Research Projects Agency of the Dept. of Defense, monitored by ONR under Contract #N00014-75-C-1091.

†Some of the results here appeared in the University of Rochester Computer Science Department TR17.

work [3] on the relation between pseudo-latin square graphs and latin squre graphs. Finally, we give extensions of these techniques to Steiner systems.

Section 1

A group throughout this paper is a Caley table. If G is a group of order n and we pick some linear ordering of G we can then view G as a binary function on {1,...,n} and the Caley table as a n x n matrix consisting of integers between 1 and n. In fact, this table is a latin square (every number between 1 and n appears exactly once in every row and in every column). On the other hand, latin squares can be viewed as binary functions; whereas functions whose multiplication tables are latin squares are called quasigroups.

Giving the definition once more, we have: A group is a binary operation * satisfying 1) and 2).

1) a) $\exists!x(a*b = x)$
   b) $\exists!x(a*x = b)$
   c) $\exists!x(x*a = b)$
2) $(a*b)*c = a*(b*c)$

A quasigroup is a binary operation satisfying 1), and a quasigroup viewed as a table or a trinary relation is a latin square.

For groups or functions it is clear what we mean by isomorphism, namely, G is isomorphic to G' if there exists a 1-1 onto function g from G to G' such that $g(x*y) = g(x)*'g(y)$. If we view G and G' as trinary relations $<\ ,\ ,\ >$ and $<\ ,\ ,\ >'$ respectively, then we get $<x,y,z>\in G$ implies $<g(x),g(y),g(z)>' \in G'$. Thus, viewing latin squares as quasigroups we say L and L' are isomorphic if there exists a permutation σ such that if we simultaneously interchange rows, columns, and values in L we get L'. But this definition is quite restrictive. We know that even independently permuting rows, columns, and values preserves the latin square properties. Thus, we say two latin squares are isotopic if we can get from one to the other by independently permuting rows, columns, and values; see [1].

Definition: Two latin squares L and L' are said to be isotopic if there exist permutations (α,β,γ) such that $<x,y,z> \in L$ implies $<\alpha(x),\beta(y),\gamma(z)>' \in L'$, which is denoted by $L \equiv L'$.

We say that two latin squares L and L' are conjugate if there exists a permutation $\alpha \in S$ such that $<x_1,x_2,x_3> \in L$ implies $<x_{\alpha(1)},x_{\alpha(2)},x_{\alpha(3)}>' \in L'$. Finally, L and L' are main class isotopic, denoted by $L \equiv_M L'$, if we can get from L to L' by a conjugation and an isotopic map.

Tarjan [9] observed that since groups of order n are generated by a set of elements of size at most log n, group isomorphism can be done in $O(n^{\log_2 n + O(1)})$ steps. Throughout the rest of this paper we shall assume that all logs are base 2. Lipton, Snyder, and Zalcstein [7], independently of Tarjan, showed a stronger result; namely, group isomorphism can be solved in $O(\log^2 n)$ space. The $O(\log^2 n)$ result seems to be dependent on the fact that groups are associative while the $O(n^{\log n + O(1)})$ result generalizes to quasigroups:

Theorem 1: Quasigroup isomorphism can be solved in $O(n^{\log n + O(1)})$ steps.

Proof: Property 1a) says the binary operation is a well-defined function. Now, 1b) and 1c) give two other well-defined functions associated with a quasigroup. We shall say that a set of elements generates the quasigroup if their closure under these three functions is the whole quasigroup. Thus, using this definition, we prove a generalization of the observation about the size of the minimal generator set.

Lemma 1: A quasigroup is generated by a set containing at most log n elements.

Proof: To prove the lemma we need only prove that if H is a proper subquasigroup of G then $|G| \geq 2|H|$. Pick $b \in G-H$. Consider the elements $H \cdot b$. Now, all the products are distinct, for if $h \cdot b = h' \cdot b$ where $h,h' \in H$ then $h = h'$ by property 1c). Secondly, $H \cdot b$ is disjoint from H for if $h = h' \cdot b$ when $h,h' \in H$ then $b \in H$ by property 1b). This contradicts the fact that $b \notin H$. Thus, $H \cdot b \subseteq G-H$ and $|H \cdot b| = |H|$ which proves the lemma.

To finish the proof of Theorem 1 we give a short description of the algorithm with two

quasigroups, G and G', as input:

1) Find a set of generators for G, containing at most log n elements, say $a_1,\ldots,a_m$.

2) For each set of m elements in G', say, $\{b_1,\ldots,b_m\}$ check to see if the map induced by $a_i \to b_i$, $1 \leq i \leq m$ is a well-defined isomorphism of G onto G'.

3) If a set of m elements of G' is found in 2) accept; otherwise reject.

Now consider isotopic latin squares. Using isotopic maps we can always put the latin square in a "normal" form; namely, the first row and first column are the sequence 1,2,...,n. This normal form is not unique. In fact, it is not unique up to isomorphism, but is almost unique up to isomorphism. Suppose that L and L' are two isotopic latin squares in normal form and (α,β,γ) is the isotopic map from L to L'. Given a permutation α, let $\alpha^{(1)}$ be the transposition $(1,\alpha^{-1}(1))$. Now the decomposition of α into $(\alpha\alpha^{(1)})(\alpha^{(1)})$ splits α into $\alpha^{(1)}$ which may move 1 while $\alpha\alpha^{(1)}$ leaves 1 fixed. The following result simply says that up to choosing who gets to be the identity L and L' are isomorphic.

Lemma 2: Given two latin squares L and L' in normal form which are isotopic by the permutations $\langle\alpha,\beta,\gamma\rangle$ then $\langle\alpha^{(1)},\beta^{(1)},\gamma^{(1)}\rangle(L)$ is isomorphic to L'.

Proof: Now, $\langle\alpha^{(1)},\beta^{(1)},\gamma^{(1)}\rangle(L) = L''$ is still isotopic to L' by $\langle\alpha' = \alpha\cdot\alpha^{(1)},\beta' = \beta\cdot\beta^{(1)},\gamma' = \gamma\cdot\gamma^{(1)}\rangle$, and L'' is in normal form. We shall show that in fact $\alpha' = \beta' = \gamma'$. Suppose that $\gamma'(V) = W$. γ' has changed the V in column V to a W. Thus, β' must move column V to column W to insure that the latin square is in normal form. Similarly, α' must move row V to row W. Therefore $\alpha' = \beta' = \gamma'$ and the lemma is proved.

Theorem 2: Isotopy of latin squares is decidable in $O(n^{\log n + O(1)})$ time.

Proof: The algorithm, on input L and L', arbitrarily puts L' in normal form and then for each of the n^2 possible candidates for the identity it puts L in normal form. Now the algorithm checks if any of these n^2 normal forms of L are in fact isomorphic to L'.

Since there are only six ways to conjugate latin squares, we get that main class isotopy is in time $O(n^{\log n + O(1)})$.

Corollary: Main class isotopy of latin squares is decidable in $O(n^{\log n + O(1)})$ time.

A Second Proof of Theorem 2: At this point we would like to make a distinction between the procedure to pick an element from a set and the procedure to guess an element in a set. To pick shall mean to arbitrarily choose some element, while to guess shall mean to try all possible elements.

Going back to the definition of isotopic latin squares, recall that we needed 3 surjective functions, say α,β,γ. We shall construct them by allowing α,β,γ to be partial one-to-one functions which preserve the trinary relations defined by the latin squares. Let the latin squares be L and L' with trinary relations $<\ ,\ ,\ >$ and $<\ ,\ ,\ >'$ respectively.

Initially,

a) pick $a,b, \in L$ (possibly the same element),

b) guess $a',b', \in L'$ (possibly the same element),

c) let $\alpha = \{(a,a')\}$, $\beta = \{(b,b')\}$, and $\gamma = \phi$.

Let $\mathrm{Dom}(\alpha)$ denote the domain of α. Given $a \in \mathrm{Dom}(\alpha)$ and $b \in \mathrm{Dom}(\beta)$ we can either extend the $\mathrm{Dom}(\gamma)$ by $(a\cdot b,\alpha(a)\cdot'\beta(b))$ [where $\cdot$ is the appropriate binary function of the three functions defined by L and similarly for $\cdot'$] or find that our guesses must be inconsistent. We shall say that α,β,γ are closed under L if domains α,β,γ are consistent with and closed under the above binary operations.

Claim: If α,β,γ are closed under L and nontrivial, then $|\mathrm{Dom}(\alpha)| = |\mathrm{Dom}(\beta)| = |\mathrm{Dom}(\gamma)|$.

Proof: By symmetry we need only show that $|\mathrm{Dom}(\beta)| \leq |\mathrm{Dom}(\gamma)|$. Pick a in $\mathrm{Dom}(\alpha)$, then $a\cdot\mathrm{Dom}(\beta) \leq \mathrm{Dom}(\gamma)$. Now all the products in $a\cdot\mathrm{Dom}(\beta)$ are distinct, therefore $|\mathrm{Dom}(\beta)| \leq |a\cdot\mathrm{Dom}(\beta)|$.

Our inductive procedure is the following:

1) Close α,β,γ under L.

2) If $L = \mathrm{Dom}(\alpha)$ then $L \cong L'$.

3) Otherwise pick $a \in L - \mathrm{Dom}(\alpha)$ and guess $a' \in L' - \mathrm{Range}(\alpha)$.

4) Set α to $\alpha \cup \{(a,a')\}$ return to 1).

By similar argument to those used in the first proof, every pass through the inductive procedure doubles the domains of α, β and γ. Thus we get an algorithm which runs in $O(n^{\log n + O(1)})$ steps.

A natural graph associated with a latin square is called a latin square graph which is defined as follows:

Definition: Given a latin square (LS), say $L(\ell_{ij})$, of size n, then the latin square graph associated with L, say G(L), has n^2 nodes g_{ij}, $1 \leq i, j \leq n$; and the nodes g_{ij} and $g_{k\ell}$ are connected if one of the following holds:

1) $i = k$
2) $j = \ell$
3) $\ell_{ij} = \ell_{k\ell}$

Latin square graphs consist of 3n n-cliques (n row cliques, n column cliques, and n value cliques). Two n-cliques are disjoint iff they are either different row, different column, or different value cliques.

Thus we get the following result:

Lemma 3: If L and L' are latin squares, and G(L) and G(L') are latin square graphs, then L is main class isotopic to L' iff G(L) is isomorphic to G(L').

If we now give an efficient method of retrieving the latin square from the latin square graph we will have an $O(n^{\log n + O(1)})$ algorithm for latin square graph isomorphism; namely, retrieve the two latin squares and check the two latin squares for main class isotopy.

Lemma 4: In $O(n^3)$ steps we can retrieve the latin square from the latin square graph where n is the dimension of the latin square.

Proof: Let G be a latin square graph on n^2 nodes. To construct a latin square we shall associate each node of G with an element in an $n \times n$ matrix $A(a_{ij})$ and also assign a value to the nodes or elements.

Algorithm:

1) Pick two connected nodes, say x_1 and x_2.
2) Find the n nodes common to x_1 and x_2. Now, n-2 of the nodes are connected to each other, say $x_3,\ldots,x_n$. Let y_2 be one of the nodes that is not connected to $x_3,\ldots,x_n$.
3) Associate a_{1j} with x_j, and set $a_{1j} = j$, $1 \leq j \leq n$.
4) Find the clique associated with x_1 and y_2, say $\{x_1,y_2,\ldots,y_n\}$. There is a unique matching between the x_i's and the y_i's.
5) Order the y_i's such that x_i is connected to y_i, for $2 \leq i \leq n$.
6) Associate a_{j1} with y_j and set $a_{j1} = j$, $2 \leq j \leq n$.
7) For each of the remaining $(n-1)^2$ nodes of G, do the following, where w is a remaining node:
   a) If w is connected to x_i then w is connected to a unique y_i and a unique x_j, $2 \leq i, j \leq n$. Set a_{ij} to 1.
   b) If w is not connected to x_i, then there exist unique integers k, i, and j such that w is connected to y_k, x_k, y_i, and y_j. Set a_{ij} to k.

Using Lemma 4 we get the following theorem.

Theorem 3: Latin square graph isomorphism is decidable in $O(n^{\log n + O(1)})$ steps.

Section 2

By our construction, it is not hard to see that an LS graph is an SR graph with parameters $(n^2,3(n-1),n,6)$. An SR graph with parameters $(n^2,3(n-1),n,6)$, for some n, is called a pseudo-latin square graph (pseudo-LS).

In his monumental paper [3], Bruck proves that large pseudo-LSGs are in fact LSGs. In particular, he shows that for $n > 23$, a pseudo-LSG is an LSG. Using his result, we get the following theorem:

Theorem 4: Isomorphism of pseudo-LS graphs can be decided in $O(n^{\log n + O(1)})$ steps.

Proof: By Lemma 4 we can in polynomial time construct the latin square from a latin square graph.

Thus, in polynomial time we can decide if a given graph is an LS graph. Hence, our algorithm will simply test if the graphs are LS graphs, and if they both are, it will use the $O(n^{\log n + O(1)})$ algorithm for deciding isomorphism. Otherwise, it will use some naive algorithm which will possibly run for exponential time. By Bruck's result, we know that for $n > 23$ the algorithm will never encounter pseudo-LS graphs which are not LS graphs. Thus, the algorithm runs in the limit for the appropriate amount of time.

Even though the theorm is theoretically and practically significant, it is not very satisfying for at least one reason. A graph with $(23)^2$ vertices is a very large graph, and no known heuristic can fill in this gap. But it should be pointed out that Bruck's result is only an upper bound on the existence of strictly pseudo-graphs, and since very few strictly pseudo-LS graphs are known, it is quite possible that Bruck's result could be strengthened.

Maximal cliques of an LS-graph can be viewed as lines and the vertices are the points. These objects are then called partial geometries or nets. We shall, following Bruck, use the term nets. Note that these nets have 3 parallel classes, one formed by the rows, one by the columns and one by the values. Such objects are called 3-nets. From [1] a net N of order n and degree k satisfies the following:

1) Each line of N contains exactly n distinct points, where $n \geq 1$,
2) Each point of N lies on exactly k distinct lines, where $k \geq 1$,
3) N has exactly kn distinct lines. These fall into k parallel classes of n lines each. Distinct lines of the same parallel class have no common points. Two lines of different classes have exactly one common point.
4) N has exactly n^2 distinct points.

By a counting argument $k \leq n+1$ and if $k = n+1$ then the k-net is called an affine plane of order n.

We shall say two nets are isomorphic if there is a surjective map over the points which preserves lines. Note that this is equivalent to saying that there exists a surjective map from lines to lines which preserves the intersections of lines. Using these definitions we get the following result.

Theorem 5: Isomorphism of k-nets is decidable in time $O(n^{\log n + O(1)})$ steps.

Proof: For $k = 0,1,2$ the pair (k,n) determines the net uniquely. Thus deciding isomorphism is trivial in these cases and we shall assume that $n+1 \geq k \geq 3$. Suppose the input is two k-nets N,N'. We simply pick 3 parallel classes from N and guess 3 parallel classes from N'. These parallel classes determine two 3-nets which we can check for isomorphism in $O(n^{\log n + O(1)})$ steps. If these 3-nets are isomorphic the procedure will in fact return all such isomorphisms from points of N to points of N'. Thus we need only check if any of these maps are isomorphisms preserving all lines of N. The 3 guesses only increase the work by $O(k^3)$ times as many steps and since $k \leq n+1$ we have proved the theorem.

The last theorem seems unsatisfactory since the information given by the extra parallel classes was not used advantageously by the procedure. In fact, the procedure was hindered by them. A more efficient approach might be to begin by guessing 3 parallel classes but if succeeding guesses determine other parallel classes then use this information to enhance the domain of the intermediate partial functions. Adding these ideas to the last theorem, we can substantially improve the running time in the special case when the net is in fact an affine plane.

Theorem 6: Isomorphism of affine planes can be decided in $O(n^{\log\log n + O(1)})$ steps.

Proof: Suppose that the input is two affine planes N and N' which we can assume are of order n. We define a procedure which inductively constructs three partial functions α,β,γ where α is from points to points and β is from lines to lines and γ is from parallel classes to parallel classes. Initially the domain of β is determined by picking 3 nonparallel lines in N and their image is determined by guessing 3 nonparallel lines in N'.

Let L be one of the parallel classes in N which is chosen during the initial step. We shall say (α,β,γ) are closed if they are closed under the following rules.

1) Two points in Dom(α) determine a line in Dom(β),
2) A line in Dom(β) determines a parallel class in Dom(γ),
3) Two nonparallel lines in Dom(β) determine a point in Dom(α),
4) A point in Dom(α) and a parallel class in Dom(γ) determine a line in Dom(β).

Note that if (α,β,γ) are closed and satisfy the initial condition then the Dom(α) and Dom(β) form, in a natural way, an affine plane.

So our inductive step in the algorithm, given (α,β,γ) closed and an initial parallel class L, is simply to pick a line in L - Dom(β), guess a line in γ(L) - Range(β) and use these choices to extend β. Finally we close (α,β,γ).

To show that this procedure runs in the stated amount of time we need only show that the domain of β is squared in size for every guess which is made. Note that each guess is made from a set of size at most n. In fact, we need only show that the Dom(β), restricted to L, is squared at each step.

Suppose that (α,β,γ) are closed, satisfying the initial condition, ℓ is the element of L-Dom(β) picked at the inductive step, and L-Dom(β) = $L_o,\ldots,L_k$ are the equivalence classes of parallel lines in Dom(β). Let ℓL_i denote points determined by ℓ and lines in L_i for $1 \leq i \leq k$ (condition 3).

Claim: The points $\ell L_1,\ldots,\ell L_{k-1}$ are distinct "new" points.

The points are all new for otherwise the points would determine ℓ by 4). Suppose that $x = \ell h = \ell h'$ where $h \in L_i$ and $h' \in L_j$ where $1 \leq i, j \leq k-1$. Now $i \neq j$ since h cannot be parallel to h'. So h and h' determine x and therefore by (3,4) ℓ is in L_o.

Since (α,β) form an affine plane the cardinality of each L_i for $0 \leq i \leq k-1$ are equal and in fact their cardinality is k. So we have generated k(k - 1) "new" points.

Claim: Each "new" point determines a distinct "new" line in L_k.

Suppose ℓh determines h' which is in L_k then the point $\ell h'$ is in Dom(α) and finally this point determines ℓ. This contradicts the fact that ℓ is not in Dom(β).

So, if L_k' is the L_k after closing L_k with respect to ℓ, then the cardinality of L_k' is equal to the square of the cardinality of L_k. Since all the L_i, $0 \leq i \leq k$, have the same cardinality, we have also squared the cardinality of L_o. Thus Theorem 6 follows.

A projective plane is simply an extension of an affine plane N which is gotten by "adding" a new point to N for each parallel class. This point is added to every line in the parallel class and all new points form a new line. And, an affine plane can be gotten from a projective plane by removing a line and all points on it.

Using this fact about the relationship between affine planes and projective planes we get:

Corollary: Isomorphism of projective planes are decidable in $O(n^{\log\log n + O(1)})$ steps.

Proof: Given two projective planes of order n, say N and N', pick a line from N and remove it to get some affine plane $\bar{N}$. Now, guess a line in N' and remove it, getting $\bar{N}'$. Using Theorem 6 we can construct all isomorphisms from $\bar{N}$ to $\bar{N}'$ in $O(n^{\log\log n + O(1)})$ steps. Using this isomorphism we can decide isomorphism of n and n'.

Following the definition of a latin square graph we define a k-net graph.

Definition: Given a k-net N of order n, where $k \leq n$, then the net-graph associated with N is a graph on n^2 vertices, one for every point in N, and where two vertices are connected if they determine a line in N.

We list a few facts and definitions from [3] about k-net graphs.

1) A k-net graph of order n is an SR graph

with parameters $(n^2, k(n-1), n-2+(k-1)(k-2), k(k-1))$.

2) Every net graph determines a unique net.

3) An SR graph with parameters as in 1) is called a pseudo-k-net-graph.

One of the fundamental theorems in [2] is:

Theorem 7 Bruck: If G is a pseudo-net-graph of order n and degree k such that $n > p(k-1)$ and $k > 1$, then G is a k-net graph, where

$$p(x) = \frac{1}{2}x^4 + x^3 + x^2 + \frac{3}{2}x \ .$$

The following is a table of tabulated values of $p(k-1)$:

k	2	3	4	5	6
p(k-1)	4	23	81	214	470

Theorem 8: If G is a k-net graph of order n and $n > (k-1)^2$, then in a polynomial in n we can reconstruct the k-net associated with G.

Proof: To prove the result, we need only show that in polynomial time we can, given two adjacent points, reconstruct the line containing them.

Let x,y be two adjacent vertices of a k-net graph G of order n. Let H be the subgraph induced by vertices common to both x and y including x and y. Now, H still contains the maximum clique containing x and y and the vertices in this clique have valence $\geq n-1$. There are $(k-1)(k-2)$ elements of H not in this clique. Each of these nonclique elements are connected to $k-1$ elements in the clique and therefore have valence at most $k-1+(k-1)(k-2)-1 = (k-1)^2 - 1$. So we can by simple valence considerations distinguish the clique elements.

Using the last two results we get the following Theorem.

Theorem 9: For fixed k pseudo-net graph isomorphism is decidable in time $O(n^{\log n + O(1)})$ steps.

Proof: Since we are only proving an O result we can assume that the order of the graphs n is $> p(k-1)$. So by Theorem 7 the graphs are in fact k-net graphs. Since $n > (k-1)^2$ we can, using Theorem 8, reconstruct the k-net. And finally, by Theorem 5, we can check the k-net for isomorphism in time $O(n^{\log n + O(1)})$ steps.

Section 3:

In this section, we will apply the techniques developed in Sections 1 and 2 to Steiner systems.

A Steiner system (V,S) with parameters (τ,κ,ν) is a partial geometry of points V and lines S (or equivalently a regular hypergraph) satisfying the following conditions:

1) there are ν points, i.e., $|V| = \nu$;

2) lines contain exactly κ points;

3) τ distinct points determine exactly one line.

A Steiner triple system is when $\tau = 2$ and $\kappa = 3$. We shall say two Steiner systems are isomorphic if there exists a surjective function from points to points which preserves lines. Finally, given a Steiner system (V,S) with parameters (τ,κ,ν), we construct a graph, called the Steiner graph $G = (X,E)$ where:

1) $X = S$,

2) $E = \{\{s_1,s_2\} | s_1,s_2 \in S \text{ and } |s_1 \cap s_2| = \kappa - 1\}$.

Using notations from above, we get:

Theorem 10: Steiner triple systems are decidable in $O(n^{\log n + O(1)})$ steps.

Proof: This follows easily by the techniques presented in Section 1.

The Steiner graph is in fact an SR graph and any graphs with the same parameter set are called pseudo-Steiner system graphs.

In [2] Bose shows that Bruck's techniques could be applied in a very general setting. In particular Bose proved that a pseudo-STS graph with strictly greater than 67 vertices must be an STS graph.

Using Bose's result we get:

Theorem 11: Isomorphism of pseudo-STS graphs is decidable in $O(n^{\log n + O(1)})$ steps.

Proof: We need only show that STS are polynomial time reconstructable from STS-graphs. Again the reconstruct follows by simple valence considerations.

References

1. Aliev, I.S. and Seiden, E. "Steiner Triple Systems and Strongly Regular Graphs," JCT 6, 1969, 33-39.
2. Bose, R.C. "Strongly Regular Graphs, Partial Geometrics and Partially Balanced Designs," Pacific J. Math. 13, 1963, 389-419.
3. Bruck, R.H. "Finite Nets, II: Uniqueness and Imbedding," Pac. J. Math. 13, 421-457.
4. Corneil, D.G. and Gotlieb "An Efficient Algorithm for Graph Isomorphisms," JACM 17, January 1970, 51-64.
5. Corneil, D.G. and Mathon, R.A. "Algorithmic Techniques for the Generation and Analysis of Strongly Regular Graphs and Other Combinatorial Configurations," to appear in Annals of Discrete Mathematics.
6. Denes, J. and Keedwell, A.D. Latin Squares and Their Applications. New York: Academic Press, 1974.
7. Lipton, R.J., Snyder, L., and Zalcstein, Y. The Complexity of Word and Isomorphism Problems for Finite Groups. John Hopkins, 1976.
8. Schmidt, D.C. and Druffel, L.E. "A Fast Backtracing Algorithm to Test Directed Graphs of Isomorphism Using Distance Matrices," JACM 23, July 1976, 433-445.
9. Tarjan, R.E., private communication.