Parallel Algorithm for Computing the Fragment Vector in Steiner Triple Systems

Erik Urland

Centre Universitaire d' Informatique
Université de Genève, 24 rue Général Dufour
1211 Genève 4, Switzerland
urland@cui.unige.ch

Abstract. In this paper we describe a linear time algorithm using $O(n^2)$ processors for computing the fragment vector in Steiner triple systems. The algorithm is designed for $SIMD$ machines having a grid interconnection network. We discuss an implementation and some experimental results obtained on the Connection Machine CM-2.

1 Introduction

Let n be a positive integer. By a *Steiner triple system* of order n, denoted by $STS(n)$, we understand a pair (V, B), where V is a set of elements called *points* (or *vertices*) such that $|V| = n$, and B is a set of such 3-subsets of V, called *lines* (*blocks* or *triples*), that every unordered pair of distinct points of V occurs exactly once among the lines of B. It is well known that $STS(n)$ exists if and only if $n \equiv 1 \ (mod\, 6)$ or $n \equiv 3 \ (mod\, 6)$. For example, an $STS(7)$ over $V = \{1, 2, ..., 7\}$ can be formed with the following set of lines $B = \{[1,2,3]; [4,1,5]; [1,6,7]; [4,6,2]; [2,5,7]; [3,4,7]; [3,5,6]\}$. Two $STS(n)$ (V_1, B_1) and (V_2, B_2) are said to be *isomorphic* if there is a bijection $\phi : V_1 \to V_2$ such that $[\alpha, \beta, \gamma] \in B_1$ if and only if $[\phi(\alpha), \phi(\beta), \phi(\gamma)] \in B_2$. A $k-line\ configuration$, $k \geq 1$, is defined as any collection of k lines of an $STS(n)$. An Erdös configuration of order k is a $k-$line configuration on $k+2$ points which contains no subconfiguration of m lines on $m+2$ points for $1 < m < k$. Two $k-$line configurations C_1 and C_2 are considered to be *isomorphic* if there is a bijection between the vertices of the configurations mapping lines to lines. If $C_1 = C_2$ then such an isomorphism is called an *automorphism*. By $frequency$ (or $number\ of\ occurrences$) of a configuration C in a given $STS(n)$ we understand the number of all different representations of the configuration C in the $STS(n)$. Let C be a configuration and let S be a subset of vertices of C. Then by a *partial* configuration P_S of C we understand a subconfiguration of C which consists only from lines having at least one vertex in S. Let C be a configuration and let π be a vertex of C. Then the $number\ of\ symmetries$ of C according to the vertex π, denoted by $\Upsilon(C, \pi)$, is the number of vertices v of C such that there is an automorphism of C mapping π to v. Let C be a configuration. Let S be a subset of vertices of C and let P_S be the partial configuration of C, which is formed by a set of lines

$L \subseteq B$. Then by $R(C, P_S)$ we denote the number of all different representations of C in $STS(n) = (V, B)$ containing L.

Non-isomorphic STSs are frequently used as source data for various kinds of statistical experiments. Similarly, we need different STSs in order to determine a linear basis for $k-$line configurations [2, 3, 5]. The classical approach is to randomly generate STSs on a computer using the hill-climbing technique [4]. This technique appears to be extremely fast but unfortunately cannot guarantee that STSs constructed in this way are non-isomorphic. As there is no known polynomial time algorithm to test isomorphism of STSs, in practice one can use invariants as a proof that two given STSs are non-isomorphic.

In this paper, we concentrate on one invariant called *fragment vector*, originally introduced by Gibbons in [1]. Consider an Erdös configuration of order 4 called the *Pasch* configuration. Let π be a point of an $STS(n)$ and let $f(\pi)$

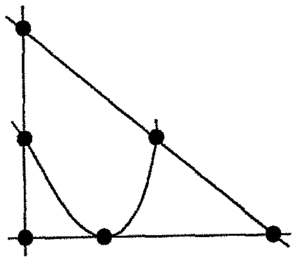

Fig. 1. Pasch configuration

denote the number of Pasch configurations containing π. Then the *fragment vector* of an $STS(n)$ is a sequence of integers $f(\pi_1), f(\pi_2), ..., f(\pi_n)$ for $\pi_i \in V$, $1 \leq i \leq n$, sorted in non-decreasing order. It is easy to see that determining the number of occurrences of the Pasch configuration in an $STS(n)$ forms the main part of the algorithm for computing the fragment vector. In the next section we shall deal with a parallel algorithm for counting the frequency of the Pasch configuration designed for $SIMD$ machines having a grid interconnection network.

2 Parallel algorithm

Before we describe the algorithm itself we shall present the following two lemmas, which will be used later in the design of the algorithm.

Lemma 1. *Let α be an arbitrary vertex of the Pasch configuration. Then* $\Upsilon(Pasch, \alpha) = 6$.

It follows from Lemma 1 that if α is a vertex of the Pasch configuration and if $S = \{\alpha\}$ then up to isomorphism, every vertex of the Pasch configuration induces the same partial configuration P_S. The following lemma shows the number of different representations of the Pasch configuration in an $STS(n)$ which can be obtained from a partial configuration P_S.

Lemma 2. *Let $STS(n)$ be a Steiner triple system of order n. Let $S = \{\alpha\}$, where α is an arbitrary vertex of the Pasch configuration and P_S denotes the partial configuration of the Pasch configuration. Then $R(Pasch, P_S) \leq 2$.*

As a next step we present an algorithm for massively parallel machine with a set of processors running in $SIMD$ mode and having a grid interconnection network. One can intuitively see that a matrix representation of STS is well suited for $SIMD - MC^2$ parallel machine. According to this representation the proposed algorithm consists of the following three levels.

Algorithm 1.

Level 1. Initialization

- For each point α of $STS(n)$, $0 \leq \alpha < n$, let $f(\alpha){:=}0$. Let $N{:=}0$, where N denotes the number of occurrences of the Pasch configuration.
- Let $T_{n\times n}$ be a matrix and let $t_{(i,j)}{:=}{-1}$ for $0 \leq i, j < n$.
- Transform the input list of triples forming $STS(n)$ to matrix $T_{n\times n}$ in such a way that the entry $t_{(i,j)}{:=}x$, $0 \leq i, j < n$, where x is a point of the triple $[i, j, x]$.
- Let $B_{n\times n}$ be an index matrix such that $b_{(i,j)}{:=}j$ for $0 \leq i, j < n$.

Level 2. Precomputation and reduction

- As each triple of $STS(n)$ is represented six times in T, we compress redundant representations. Let $A_{n\times n}$ be a matrix and let $a_{(i,j)}{:=}t_{(i,j)}$, $0 \leq i, j < n$. For all rows i, $0 \leq i < n$, and each column j, $0 \leq j < n$, if $a_{(i,j)} \neq -1$ then $a_{(i,a_{(i,j)})}{:=}{-1}$.
- Delete the entries $a_{(i,j)} < 0$, $0 \leq i, j < n$, from the matrix A and the corresponding entries $b_{(i,j)}$ from B. Note that after this reduction the matrices A and B are of the size $n \times \varphi$, where $\varphi = \frac{n-1}{2}$.
- Let $C_{n\times\varphi}$ and $D_{n\times\varphi}$ be two matrices and let $c_{(i,j)}{:=}a_{(i,j)}$ and $d_{(i,j)}{:=}b_{(i,j)}$ for $0 \leq i < n$ and $0 \leq j < \varphi$.

Level 3. Main computation

- Let us shift the entries $c_{(i,j)}$ and $d_{(i,j)}$ of the matrices C and D in such a way that $c_{(i,j)}{:=}c_{(i,j+1)}$ and $d_{(i,j)}{:=}d_{(i,j+1)}$ for $0 \leq i < n$ and $0 \leq j < \varphi - 1$.
- Delete the last column of the matrices A, B, C and D, and set $\varphi{:=}\varphi - 1$. Assume that vertex $\alpha \in P_{\{\alpha\}}$. Then the triple $h_1{=}[i, b_{(i,j)}, a_{(i,j)}]$ corresponds to one line of $P_{\{\alpha\}}$, for $\alpha = i$ and for some j-th triple from the list of triples containing α. Similarly, the triple $h_2{=}[i, d_{(i,j)}, c_{(i,j)}]$ corresponds to the second line of $P_{\{\alpha\}}$. Thus the pairs of entries $\{a_{(i,0)}, b_{(i,0)}\}$, $\{a_{(i,1)}, b_{(i,1)}\}$,

..., $\{a_{(i,\varphi-1)}, b_{(i,\varphi-1)}\}$ of the matrices A and B, together with some vertex $\alpha = i$, form the triples which represent one line of the partial configuration $P_{\{\alpha\}}$. The second line of $P_{\{\alpha\}}$ is represented in a similar way, by pairs of corresponding entries of the rows i in C and D. Note that the lines h_1 and h_2, containing the j-th pair of entries of the matrices A, B and C, D, forming $P_{\{\alpha\}}$ for $\alpha = i$, $0 \leq \alpha < n$, cannot be the same.

- For all $a_{(i,j)}$, $b_{(i,j)}$, $c_{(i,j)}$ and $d_{(i,j)}$, $0 \leq i < n$ and $0 \leq j < \varphi$, check if $t_{(a_{(i,j)},c_{(i,j)})} = t_{(b_{(i,j)},d_{(i,j)})}$. If yes, then without loss of generality the two other lines, not forming $P_{\{\alpha\}}$ for $\alpha = i$, have a common point. Thus we obtain one representation of the Pasch configuration from Lemma 2. In this case let $N{:=}N{+}1$ and $f(\pi){:=}f(\pi){+}1$ for all points π forming the triples which represent the Pasch configuration.
- Repeat the previous step for all $a_{(i,j)}$, $b_{(i,j)}$, $c_{(i,j)}$ and $d_{(i,j)}$, $0 \leq i < n$ and $0 \leq j < \varphi$, under the condition $t_{(a_{(i,j)},d_{(i,j)})} = t_{(b_{(i,j)},c_{(i,j)})}$, which checks the second representation of the Pasch configuration from Lemma 2.
- Repeat the last four steps until $\varphi = 1$.
- By Lemma 1, let $N{:=}N/6$ and for each point α of $STS(n)$, $0 \leq \alpha < n$, let $f(\alpha){:=}f(\alpha)/6$. The computation is performed and the algorithm terminates.

Theorem 3. *Algorithm 1 computes the frequency of the Pasch configuration in an $STS(n)$ in linear time using $O(n^2)$ processors.*

3 Implementation and experimental results

As the processors of the Connection Machine CM-2 can be configured as a k-dimensional grid, we use this computational model for implementing our algorithm. The speedup of the parallel approach becomes significant with the order of STS greater than 100. For $STS(249)$ we have achieved speedup approximately 30 compared to the best known $O(n^3)$ sequential algorithm running on a Sun SPARCstation 4 computer. Note that the original algorithm can be optimized by assuming wrap-around connections among processors in the grid, but this modification gives only slightly better results.

References

1. Gibbons, P. D.: *Computing techniques for the construction and analysis of block designs*, Ph.D. Thesis, University of Toronto, 1976.
2. Grannell, M. J.-Griggs, T. S. and Mendelsohn, E.: *A small basis for four-line configurations in Steiner triple systems*, Journal of Combinatorial Designs, Vol. 3, No. 1 (1995), p. 51-59.
3. Horak, P.-Phillips, N.-Wallis, W. D. and Yucas, J.: *Counting frequencies of configurations in Steiner triple systems*, to appear in Journal of Combinatorial Designs.
4. Stinson, D. R.: *Hill-climbing algorithms for the construction of combinatorial designs*, Annals of Discrete Math. **26**, (1985), p. 321-334.
5. Urland, E.: *A linear basis for the 7-line configurations*, submitted for publication, Journal of Comb. Math. and Comb. Computing.