BIT 29 (1989), 448–456

THE AVERAGE PERFORMANCE OF A PARALLEL STABLE MARRIAGE ALGORITHM

S. S. TSENG

Institute of Information Science, National Chiao Tung University, Hsinchu, Taiwan 30050, Republic of China

Abstract.

In this paper, Tseng and Lee's parallel algorithm to solve the stable marriage prolem is analysed. It is shown that the average number of parallel proposals of the algorithm is of order n by using n processors on a CREW PRAM, where each parallel proposal requires $O(\log\log(n))$ time on CREW PRAM by applying the parallel selection algorithms of Valiant or Shiloach and Vishkin. Therefore, our parallel algorithm requires $O(n\log\log(n))$ time. The speed-up achieved is $\log(n)/\log\log(n)$ since the average number of proposals required by applying McVitie and Wilson's algorithm to solve the stable marriage problem is $O(n\log(n))$.

CR Category: F.2.0

Keywords: average case analysis, parallel algorithm, stable marriage.

1. Introduction.

The stable marriage problem was first introduced and solved by Gale and Shapley [2]. Later, several stable marriage algorithms [6, 7, 8, 13, 1] were proposed to solve the problem. In [13] Wilson showed that the worst case performance and average performance of the stable marriage algorithm [6] are $n^2 - n + 1$ and $O(n\log(n))$, respectively. Recently, two parallel stable marriage algorithms have been presented [3, 11]. Tseng and Lee's algorithm, which is a parallel version of McVitie and Wilson's algorithm, has worst case performance $n^2 - 2n + \log n$ which is no faster than that of McVitie and Wilson's. In [9] and [4] it was suggested that parallel stable marriage algorithms cannot be expected to provide high speed-up on the average. We shall show in this paper that the average number of parallel proposals of Tseng and Lee's algorithm [11] is $O(n)$. The result is obtained by using n processors on a CREW PRAM, which is a shared-memory computer allowing in a single cycle the processors to perform concurrent reads from the same location but not concurrent writes (Concurrent Read Exclusive Write Parallel Random Access

Received August 1988. Revised February 1989.

Machine). Since each parallel proposal requires $O(\log\log(n))$ time on a CREW PRAM by applying either of the parallel selection algorithms proposed in [12] or in [10], the speed-up achieved is $O(\log(n)/\log\log(n))$.

2. Tseng and Lee's parallel stable marriage algorithm

The stable marriage problem is defined as follows. We are given n men and n women and each man ranks each woman from 1 to n and vice versa. A set of marriages is a one-to-one correspondence of men to women. If there does not exist an unmarried pair both preferring each to their current partners, then we say that this set of marriages is stable.

Tseng and Lee's parallel algorithm [11] is based upon the divide-and-conquer approach. We first divide the problem into two sub-problems, by halving the male ranking matrix and then recursively applying the algorithm to find the male optimal stable solutions for these two subproblems. We then merge these two solutions.

Tseng and Lee's algorithm

Input: A male ranking matrix.
Output: A male optimal stable solution.
Step 1: Divide the problem into two sub-problems, by halving the male ranking matrix. Call these two sub-problems P_1 and P_2.
Step 2: Recursively apply this algorithm to find optimal stable solutions for P_1 and P_2. Call these two solutions S_1 and S_2.
Step 3: Apply Algorithm B which is a merging algorithm to combine S_1 and S_2 into S.

In a solution S, let W_s denote the set of women who are proposed to by more than one man. Assume $M_{i_1},\ldots,M_{i_k}$, $k \geq 2$, propose to the same woman W_i, where $W_i \in W_s$. Without loss of generality, we shall assume that so far as W_i is concerned, the ranking of M_{i_k} is the highest. Then W_i will accept M_{i_k} and reject $M_{i_1},\ldots,M_{i_{k-1}}$. We shall say that $\{M_{i_1},\ldots,M_{i_{k-1}}\}$ is the set of rejected men of W_i. Let R_s be the union of the sets of rejected men of members in W_s. Then R_s is called the rejected men associated with S.

Algorithm B

Input: Two male optimal stable solutions S_1 and S_2 and their associated ranking matrices.

Output: A male optimal stable solution which combines S_1 and S_2.

Step 1: Let S be the union of S_1 and S_2.

Step 2: If no two men propose to the same woman in S, then accept S as the solution and return. Otherwise, go to Step 3.

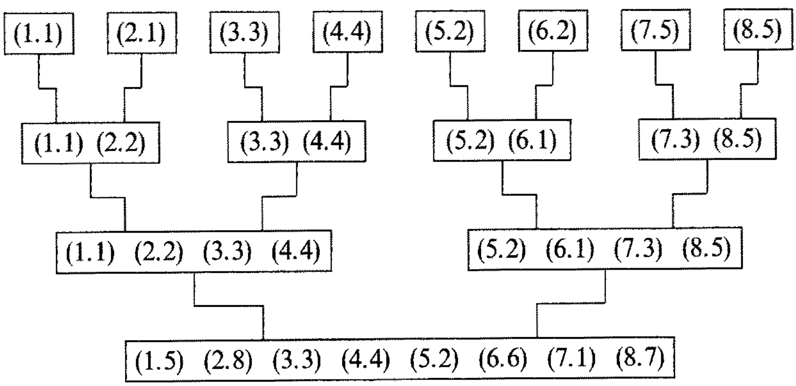
Step 3: For each man M_i in R_s, and for the set of rejected men associated with S, replace (M_i, W_j) in S by (M_i, W_k) where W_k is the next best choice of M_i.

Step 4: Go to Step 2.

Let us consider the following example:

M_1	1	4	5	6	2	3	7	8	W_1	8	7	6	4	3	5	1	2
M_2	1	2	4	8	7	6	3	5	W_2	1	5	4	3	6	2	7	8
M_3	3	2	1	4	8	7	5	6	W_3	1	3	2	4	7	8	6	5
M_4	4	6	7	8	1	3	2	5	W_4	4	7	8	2	6	1	3	5
M_5	2	3	4	6	7	1	8	5	W_5	3	1	2	5	6	8	7	4
M_6	2	1	8	6	5	7	3	4	W_6	1	2	3	4	5	6	7	8
M_7	5	3	1	4	8	6	2	7	W_7	2	1	3	5	4	6	7	8
M_8	5	7	6	3	1	2	4	8	W_8	7	4	2	1	5	6	8	3
	Men's Ranking									Women's Ranking							

The merging steps are illustrated as follows:



Finally, we obtain the male optimal stable solution:
(1,5) (2,8) (3,3) (4,4) (5,2) (6,6) (7,1) (8,7).

3. Analysis of Tseng and Lee's algorithm

We are concerned with finding the average number of parallel proposals, which is required to obtain the male optimal stable solution by applying Tseng and Lee's algorithm [11]. In our model n proposals can be performed simultaneously and each parallel proposal requires $O(\log\log n)$ time by applying either of the parallel selection algorithms proposed by Valiant [12] or Shiloach and Vishkin [10]. Assume that there are n men and n women. Define $A(n)$ to be the average number of parallel proposals required to obtain the male optimal stable solution by applying the algorithm. It is obvious that $A(1) = 0$. Then we define $M(n/2^k)$ to be the average number of parallel proposals required to merge two male optimal stable solutions, where each solution consists of $n/2^k$ men. Applying Tseng and Lee's algorithm, the stable marriage solution of $n/2^k$ men can be obtained by merging two solutions of $n/2^{k+1}$ men. Without loss of generality, assume $n = 2^{\alpha+1}$ where α is an integer. We have

$$
\begin{aligned}
&A(1) = 0\\
&A(n/2^i) = A(n/2^{i+1}) + M(n/2^{i+1})\\
\text{Thus }\ &A(n) = A(n/2) + M(n/2)\\
&\quad = A(n/4) + M(n/4) + M(n/2)\\
&\quad = \dots\\
&\quad = A(1) + \sum_{i=0}^{\alpha} M(2^i), \quad \text{where } \alpha = \log_2(n/2).
\end{aligned}
$$

In the process of merging two male optimal stable solutions S_1 and S_2 of i men, it can be seen that in both S_1 and S_2 there are $\binom{n}{i}$ possibilities to choose i women from n women. Assume that each of the $\binom{n}{i}$ is equally likely. So the total number of input possibilities of S_1 and S_2 is $\binom{n}{i}\binom{n}{i}$. If there are j men in S_2 who propose to the same women ($|R_s| = j$) in the two male optimal stable solutions of i men, then there are $\binom{i}{j}$ possible ways to choose the j rejected men. Once the j rejected men are fixed, there are $\binom{n-i}{i-j}$ different ways to choose the other $i-j$ men.

Let P_{ij} be the probability that there are j conflict pairs of men who propose to the same woman in the two male optimal stable solutions of i men. Therefore, we have

$$
P_{i1} = \binom{n}{i}\binom{i}{1}\binom{n-i}{i-1}\Big/\binom{n}{i}^2, \quad P_{i2} = \binom{n}{i}\binom{i}{2}\binom{n-i}{i-2}\Big/\binom{n}{i}^2
$$

and

$$
P_{ij} = \binom{n}{i}\binom{i}{j}\binom{n-i}{i-j}\Big/\binom{n}{i}^2 = \binom{n-i}{i-j}\binom{i}{j}\Big/\binom{n}{i}
$$

So $M(i) = \sum_{j=1}^{i} P_{ij}Q_{ij}$ is defined as the average number of parallel proposals required to merge two male optimal stable solutions of j conflicts within $2i$ men. At the merging stage, we assume that the probability of proposing to a proposed woman is $(2i-j-1)/(n-1)$ and that to an unproposed woman is $(n-2i+j)/(n-1)$. To simplify our notation, let $m=n-1$ and $f_i(j)=(2i-j-1)/m$. After the execution of one parallel proposal in the merging stage, there are $\binom{j}{k} f_i(j)^{j-k}(1-f_i(j))^k$ possibilities that k of the j rejected men will be engaged. Therefore we have

$$(1)\qquad Q_{ij}=1+\sum_{k=0}^{j}\binom{j}{k} f_i(j)^{j-k}(1-f_i(j))^k Q_{i,j-k}\quad\text{for}\quad 1\le j\le i\le n/2$$

and $Q_{i0}=1$, for $1\le i\le n/2$.

Let $g_i(x)=1/(1-f_i(x)^x)$. Then

$$(2)\qquad Q_{ij}=g_i(j)+g_i(j)\sum_{k=1}^{j}\binom{j}{k} f_i(j)^{j-k}(1-f_i(j))^k Q_{i,j-k}\quad\text{for}\quad j\ge 1.$$

Observe that the first derivative of $g_i(x)$ is negative for $x\in[1,i]$ and $i\le n/2$. By this property, we have the following lemma.

LEMMA 1: *$g_i(x)$ is a decreasing function for $x\in[1,i]$, $i\le n/2$.*

PROOF. We only have to show that the first derivative of $g_i(x)$ is negative.

$$\begin{aligned} g_i'(x) &= -(1-f_i(x)^x)'/(1-f_i(x)^x)^2 \\ &= -\left(-f_i(x)^x\left(\frac{-x}{mf_i(x)}+\ln(f_i(x))\right)\right)\Big/(1-f_i(x)^x)^2 \end{aligned}$$

Since $\ln(f_i(x))<0$ for $x\in[1,i]$ and $i\le n/2$, we find $g_i'(x)<0$. ■

The next two lemmas are used to derive the upper bounds of $g_{n/2}(j)$ for $1\le j<m^{1/2}$ and $m\le j\le n/2$, respectively.

LEMMA 2: *If $1\le j<m^{1/2}$ then $g_{n/2}(j)<2m/j^2$.*

PROOF.

$$\begin{aligned} 2m/j^2-g_{n/2}(j) &= 2m/j^2-1/(1-f_{n/2}(j)^j) \\ &= \frac{2m(m^j-(m-j)^j)-j^2m^j}{j^2(m^j-(m-j)^j)} \\ &= \frac{2m\left(m^j-\sum_{k=0}^{j}\binom{j}{k}m^{j-k}(-j)^k\right)-j^2m^j}{j^2(m^j-(m-j)^j)} \end{aligned}$$

$$\geq \frac{m^j j^2 - j(j-1)m^{j-1}j^2}{j^2(m^j-(m-j)^j)}$$

$$+ \frac{2m\sum_{k=2}^{\lfloor j/2 \rfloor}\left(\binom{j}{2k-1}m^{j-2k+1}j^{2k-1} - \binom{j}{2k}m^{j-k}j^{2k}\right)}{j^2(m^j-(m-j)^j)}$$

If $j < m^{1/2}$ then the following formula is true:

$$\binom{j}{2k-1}m^{j-2k+1}j^{2k-1} - \binom{j}{2k}m^{j-2k}j^{2k}$$

$$= m^{j-2k}j^{2k-1}j!\frac{2km-(j-2k+1)j}{(2k)!(j-2k+1)!} > 0.$$

Therefore $2m/j^2 - g_{n/2}(j) > \dfrac{m^j j^2 - j(j-1)m^{j-1}j^2}{j^2(m^j-(m-j)^j)} > 0.$ ■

LEMMA 3: *If $m \leq j \leq n/2$, then $g_{n/2}(j) \leq C$ where C is a constant.*

PROOF. From [5] we have $(n+a)^{n+b} = n^{n+b}e^a(1 + a(b-a/2)/n + O(n^{-2}))$. With $m^{1/2} = p$ this implies that $g_{n/2}(p) = p^p/(p^p-(p-1)^p) = e/(e-1+1/(2p) + O(m^{-1})) \leq C$. According to Lemma 1, $g_{n/2}(j) \leq g_{n/2}(p) \leq C$, for $p \leq j \leq n/2$. ■

The next lemma shows the upper bound of $g_i(j)$ for $1 \leq j \leq i \leq n/4$.

LEMMA 4. *$g_i(j) \leq 2$ for $1 \leq j \leq i \leq n/4$.*

PROOF. Then $g_i(j) = 1/(1-f_i(j)^j) = \left[1 - \left(\dfrac{2i-j-1}{m}\right)^i\right]^{-1}$. Because $\left(\dfrac{2i-j-1}{m}\right)^j \leq \dfrac{2i-j-1}{m}$ for $1 \leq j \leq i \leq n/4$, we get
$g_i(j) \leq 1/(1-(2i-j-1)/m) = m/(m+2i+j+1) \leq 2$. ■.

Let $h(i) = \sum_{k=1}^{i} 2m/k^2$. Before deriving the order of $M(i)$, we need to know the behavior of Q_{ij}.

LEMMA 5: *$Q_{n/2,j} < h(j) + nC/2$, for $1 \leq j \leq n/2$.*

PROOF. From (2),

$$Q_{n/2,j} = g_{n/2}(j) + g_{n/2}(j)\sum_{k=1}^{j}\binom{j}{k}f_{n/2}(j)^{j-k}(1-f_{n/2}(j))^k Q_{i,j-k}$$

$$= g_{n/2}(j) + g_{n/2}(j)/m^j\sum_{k=1}^{j}(m-j)^{j-k}j^k Q_{i,j-k}.$$

The lemma is obviously true when $n = 2$. If $n > 2$ and $j = 1$ then $Q_{n/2,j} = g_{n/2}(1) + Q_{n/2,0} = m + 1 < \mathrm{h}(1)$. Assume that $Q_{n/2,j-1} < h(j-1)$ for $j < j' \leq m^{1/2}$.

Then

$$Q_{n/2,j'} = g_{n/2}(j') + (g_{n/2}(j')/m^{j'}) \sum_{k=1}^{j'} \binom{j'}{k} (m-j')^{j'-k} j'^k Q_{n/2,j'-k}$$

$$= g_{n/2}(j') + (g_{n/2}(j')/m^{j'}) \sum_{k=1}^{j'} \binom{j'}{k} (m-j')^{j'-k} j'^k Q_{n/2,j'-k}$$

$$- g_{n/2}(j')h(j'-1)((m-j')/m)^{j'} + g_{n/2}(j')h(j'-1)((m-j')/m)^{j'}$$

$$< g_{n/2}(j') + (g_{n+2}(j')h(j'-1)/m^{j'}) \sum_{k=0}^{j'} \binom{j'}{k} (m-j')^{j'-k} j'^k$$

$$- g_{n/2}(j')h(j'-1)((m-j')/m)^{j'}.$$

But $\sum_{k=0}^{j'} \binom{j'}{k} (m-j')^{j-k} j'^k = (m - j' + j')^{j'} = m^{j'}$. So $Q_{n/2,j} < g_{n/2}(j') + g_{n/2}(j')h(j'-1)(1 - ((m-j')/m)^{j'}) = g_{n/2}(j') + h(j'-1)$.

According to Lemma 2, $g_{n/2}(j) < 2m/j^2$, for $j < m$, and hence $Q_{n/2,j} < 2m/j'^2 + h(j'-1) = h(j')$.

From Lemma 3, it can easily be proved that $Q_{n/2,j} < h(j) + nC/2$, for $1 \leq j \leq n/2$, by applying the same argument. ■

LEMMA 6: $Q_{ij} \leq 3j$, *for* $i \leq n/4$.

PROOF. From (2),

$$Q_{ij} = g_i(j) + g_i(j) + \sum_{k=1}^{j} \binom{j}{k} f_i(j)^{j-k}(1 - f_i(j))^k Q_{i,j-k}.$$

According to Lemma 4, $Q_{ij} = g_i(1) + Q_{i,0} \leq 3$. Assume that $Q_{i,j-1} \leq 3(j-1)$ for $j < j' \leq i \leq n/4$. Then

$$Q_{ij'} = g_i(j') + g_i(j') \sum_{k=1}^{j'} \binom{j'}{k} f_i(j')^{j'-k}(1 - f_i(j'))^k Q_{i,j'-k}$$

$$- g_i(j')f_i(j')^{j'} Q_{i,j'-1} + g_i(j')f_i(j')^{j'} Q_{i,j'-1}$$

$$\leq g_i(j') - g_i(j')f_i(j')^{j'} Q_{i,j'-1}$$

$$+ 3(j'-1)g_i(j') \sum_{k=0}^{j'} \binom{j'}{k} f_i(j')^{j'-k}(1 - f_i(j'))^k$$

$$\leq g_i(j') - 3(j'-1)g_i(j')(1 - f_i(j')^{j'})$$

$$\leq 2 + 3(j'-1) \leq 3j'.$$

Thus we conclude that $Q_{ij} \leq 3j$, for $i \leq n/4$. ■

Theorem 1 now follows in a straight forward manner.

THEOREM 1: *The average performance of Tseng and Lee's parallel stable marriage algorithm is $O(n)$ by using n processors.*

PROOF. According to Lemma 5, we have

$$Q_{n/2,j} < h(j) + nC/2 = 2m \sum_{k=1}^{j} (1/k)^2 + nC/2 \leq 4n + nC/2.$$

$$\begin{aligned} M(n/2) &= \sum_{j=1}^{n/2} P_{n/2,j} Q_{n/2,j} \\ &= \binom{n}{n/2}^{-1} \sum_{j=1}^{n/2} \binom{n-n/2}{n/2-j}\binom{n/2}{j} Q_{n/2,j} \\ &\leq (4 + C/2)n \binom{n}{n/2}^{-1} \sum_{j=1}^{n/2} \binom{n-n/2}{n/2-j}\binom{n/2}{j}. \end{aligned}$$

From the formula

$\sum_k \binom{r}{k}\binom{s}{n-k} = \binom{r+s}{n}$ it follows that $M(n/2) \leq (4 + C/2)n$. According to Lemma 6, we have

$$\begin{aligned} M(i) &\leq \binom{n}{i}^{-1} \sum_{j=1}^{i} \binom{n-i}{i-j}\binom{i}{j} 3j \\ &= 3i \binom{n}{i}^{-1} \sum_{j=1}^{i} \binom{n-i}{i-j}\binom{i-1}{j-1} \\ &= 3i \binom{n-1}{i-1} \Big/ \binom{n}{i} < 3i, \text{ for } i \leq n/4. \end{aligned}$$

Thus with $\alpha = \log_2(n/2)$:

$$\begin{aligned} A(n) &= A(1) + \sum_{i=0}^{\alpha} M(2^i) \\ &< 3 \sum_{i=1}^{\alpha-1} 2^i + (4 + C/2)n = O(n). \qquad \blacksquare \end{aligned}$$

4. Concluding remarks.

We have shown that the average number of parallel proposals of Tseng and Lee's algorithm is $O(n)$ by using n processors on a CREW PRAM, where each parallel proposal requires $O(\log\log(n))$ time. The speed-up of the algorithm is

$O(\log n/\log\log n)$ since the average sequential time is $O(n \log n)$. This research leads us to an interesting problem: Is there a parallel stable marriage algorithm which uses $O(n)$ processors and runs in $O(\log n)$ time? It might be worthwhile to conduct future research in this direction.

Acknowledgements.

The author is grateful to an anonymous referee for his valuable comments.

REFERENCES

1. Allison, L., *Stable marriages by coroutines*, Information Processing Letters, Vol. 16, No. 2, Feb. 1983, pp. 611–65.
2. Gale, D. and Shapley, L. S., *College admissions and the stability of marriage*, Am. Math. Monthly 69, 1962, pp. 9–15.
3. Hull, M. E. C., *A parallel view of stable marriages*, Information Processing Letters, Vol. 18, No. 1, Feb. 1984, pp. 63–66.
4. Kapur, D. and Krishnamoorthy, M. S., *Worst-case choice for the stable marriage problem*, Information Processing Letters, Vol. 21, July 1985, pp. 27–30.
5. Knuth, D. E., *The Art of Computer Programming*, Vol. 1/Fundamental Algorithms, 2nd Edition, Addison-Wesley, Reading, Mass., 1973.
6. McVitie, D. G. and Wilson, L. B., *Stable marriage assignment for unequal sets*, BIT 10, 1970, pp. 295–309.
7. McVitie, D. G. and Wilson, L. B., *The stable marriage problem*, Communication of the ACM, Vol. 14, No. 7, July 1971, pp. 486–490.
8. McVitie, D. G. and Wilson, L. B., *Algorithm 411, Pts. 1–3. Three procedures for the stable marriage problem*, Communication of the ACM, Vol. 14, No. 7, July 1971, pp. 491–492.
9. Quinn, M. J., *A note on two parallel algorithms to solve the stable marriage problem*, BIT 25, 1985, pp. 473–476.
10. Shiloach, Y. and Vishkin, U., *Finding the maximum, merging, and sorting in a parallel computation model*, Journal of Algorithms, Vol. 2, 1981, pp. 88–102.
11. Tseng, S. S. and Lee, R. C. T., *A parallel algorithm to solve the stable marriage problem*. BIT 24, 1984, 308–316.
12. Valiant, L. G., *Parallelism in comparison problems*, SIAM Journal of Computing, Vol. 4, No. 3, Sep. 1975, pp. 348–455.
13. Wilson, L. B., *An analysis of the stable marriage assignment algorithm*. BIT 12, 1972, pp. 569–575.