Parallelizing the Stable Matching Problem

Alec Kyritsis, Scott Wynn, Stephora Cesar-Alberi, Mentor: Dr. Enyue Lu
Research Experience for Undergraduates, National Science Foundation

Abstract

Sequential algorithms for the Stable Matching Problem are often too slow in the context of some large scale applications like switch scheduling. Parallel architectures can offer a notable decrease in complexity. We propose a stable matching algorithm using n^2 processors that converges in *O(nlog(n))* average runtime. The algorithm is structurally based on the Parallel Iterative Improvement (PII) Algorithm, which successfully finds a stable matching in approximately 90% of cases. We suggest alternative selection method for pairs in the PII algorithm, called Dynamic Selection, leading to an algorithm that fully converges over 200,000 trials and generally converges much faster. We also suggest the NM1-Right selection method, which converges in approximately 99.9% of cases and can be combined with Colin White's PII-SC algorithm's cycle detection and Gale-Shapley Initiation to remove all observed cycles.

Problem Introduction

The Stable Matching Problem(SMP) may be presented through the analogy of marriage: Given *n* men and *n* women each with a ranking of preference for the opposite sex, we hope to identify a matching such that no man and no woman prefers each other to their current partner (figure 1). If no such pair exists, then the matching is said to be *stable*. Otherwise, the matching is *unstable*. The problem has numerous applications in social, physical and computational science. Of particular interest is switch scheduling given the time constraint.

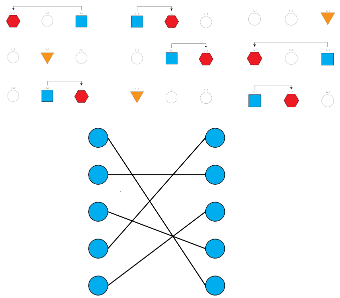

Figure 1: *Top*: An instance of cycling. Unstable Pairs in squares and blocking pairs in hexagons. Arrows indicate blocking pairs satisfied by algorithm. NM2 pairs as a result of selection type in triangles. *Bottom*: Matching on bipartite graph.

PII Overview

The SMP may be represented on an *n x n* matrix where each entry $p_{i,j}$ corresponds to the pairing of man *i* with women *j*. For an arbitrary pair, the *left value* denotes man *i*'s preference for women *j* (resp. *right value*). We call a $p_{m,l}$ a *blocking pair* if man *m* and woman *l* prefer each other to their current mates. The PII algorithm exploits this matrix representation, leveraging an *n x n* grid of processors to find blocking pairs. The algorithm is divided into two phases: Initiation and Iteration. The initiation phase randomly generates an initial matching, M_0. The iteration phase constructs a sequence of matchings, $M_1, \dots, M_k$ such that M_{i+1} contains the most unstable blocking pairs of M_i. This is achieved as follows: For unstable matching M_i Identify all blocking pairs. For each row, select the blocking pair with the minimal left value. If multiple blocking pairs occur in the same column, take the one with the minimal right value. We term this set *NM1 pairs.* For those men and women left unmatched, the algorithm deterministically pairs them under the new matching (Figure 2). However, this approach may generate a sequence of matchings $M_k, M_{k+1}, \dots, M_{k+i}$ where $M_k = M_{k+i}$. We call this a cycle (Figure 1).

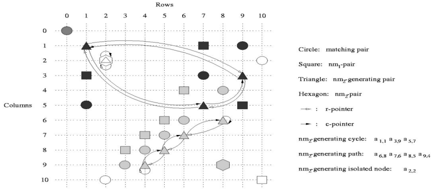

Figure 2: Iteration phase of the original PII Algorithm provided by [3].

NM1-Right Selection

NM1-Right Selection modifies the original greedy selection method. Rather than take all blocking pairs across a row, we only consider those where the right value decreases relative to the old matching pair (Figure 3). Selection then proceeds as usual. In the event that the matching is unstable but no blocking pairs may be chosen, we return to normal NM1 selection. This method is deployed after *n* iterations and continues until *3n* iterations. We have proven that when not supplemented with NM1 selection, this method may not cycle (Proven Results).

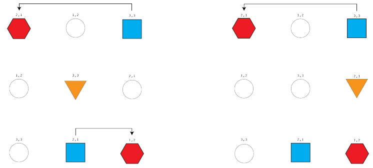

Figure 3: Left: NM1 Selection. Right: NM1-Right Selection.

Dynamic Selection

Dynamic selection choses NM1 pairs similar to the original PII Algorithm. However, for each NM1-generating pair $p_{i,j}$ selected in row *i*, we check if its left value is smaller than the minimum left value of all NM1 pairs previously selected from that row; call the pair that corresponds to this value $p_{i,l}$. If this is the case, we take $p_{i,j}$ as an NM1 pair. Otherwise, we take $p_{i,l}$ as an NM1 pair (Figure 4). In practice, we impose a *wait condition* such that if a comparison occurs at iteration *k*, we may not compare again until iteration *k + W* (where *W* is an integer value) or a new minimum is found. This method is deployed after *3n* iterations.

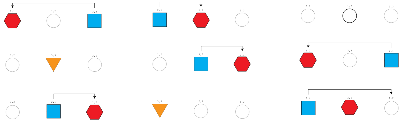

Figure 4: Rightmost selection results in stable matching.

Proven Results

Our primary theoretical results concern NM1-Right Selection and are as follows:

- **NM1-Cycle Freeness Lemma:** Let A be an instance of the PII algorithm. If A uses Right-Minimum Selection, then A is NM1-cycle free.
- **Path Shortening Lemma:** Let P be an NM1-path taken by the Algorithm. Then each time P is traversed, it's length decreases by at least two.
- **Cycle Freeness Theorem:** Let A be an instance of the PII algorithm. If A uses Right-Minimum Selection, then A is cycle free.

Results

NM1-Right

Input Size (N)	Random Initialization		GS Initialization	
	Mean Iterations to Convergence	Cycles	Mean Iterations to Convergence	Cycles
50	21.02	2	19.978	1
60	23.973	1	22.903	0
70	28.545	4	25.663	0
80	33.507	1	28.609	0
90	35.927	0	31.644	0
100	40.197	1	35.254	0

NM1-Right & Dynamic Select

Input Size (N)	Random Initialization		GS Initialization	
	Mean Iterations to Converge	Cycles	Means Iteration to Convergence	Cycles
10	3.986	0	3.1951	0
20	8.2391	0	7.7714	0
30	12.6825	0	11.9154	0
40	17.8182	0	16.026	0
50	22.1112	0	19.7024	0
60	26.8328	0	23.2906	0
70	31.3267	0	26.8111	0
80	34.6355	0	29.6866	0
90	39.1537	0	33.0558	0
100	41.5797	0	35.9822	0

Conclusion

Our findings suggest that for greedy improvement algorithms aimed at solving the SMP, optimized selection methods can hasten convergence and mitigate cycling. A phased selection approach appears to yield best results, with no cycling and faster average convergence observed in over 200,000 trials regardless of initialization method. Our algorithm also lends theoretical insight into one notable cause of cycling; namely, blocking pair selection from two separate stable marriages.

References & Acknowledgements

This work is proudly supported by NSF grant CNC-2149591.

[1] Enrico Maria Fenoaltea, Izat B Baybusinov, Jianyang Zhao, Lei Zhou, and Yi-Cheng Zhang. The stable marriage problem: an interdisciplinary review from the physicist's perspective. Physics Reports, 917:1–79, 2021.

[2] David Gale and Lloyd S Shapley. College admissions and the stability of marriage. The American Mathematical Monthly, 69(1):9–15, 1962.

[3] Enyue Lu, Mei Yang, Yi Zhang, and SQ Zheng. Design and implementation of an acyclic stable matching scheduler. In GLOBECOM'03. IEEE Global Telecommunications Conference (IEEE Cat. No. 03CH37489), volume 7, pages 3938–3942. IEEE, 2003.

[4] Yuxiang Zhang, Lin Cui, and Yuan Zhang. A stable matching based elephant flow scheduling algorithm in data center networks. Computer Networks, 120:186–197, 2017.