A Parallel Approximation Hitting Set Algorithm for Gene Expression Analysis

D. P. Ruchkys
Universidade de São Paulo

S. W. Song
Universidade de São Paulo

Gene Expression Analysis

- Given an experiment where expression levels of thousands of genes are measures.
- We consider the problem of determining which genes affect the expression level of a given gene.

Our Problem

- Given an experiment with n genes of a set $E = \{a_0, a_1, ..., a_{n-1}\}$ whose expression levels are measured in a time series of m measures (typically $n >> m$). We have a total of nm values of 0's or 1's.
- Our algorithm (based on Ideker *et al.* [ITK00]) receives an $m \times n$ matrix of such values and determine, for a given gene a_{n-1}, which other genes are responsible for the expression level of a_{n-1}.
- Example.

$$M = \left|\begin{array}{cccc|c} x_0 & x_1 & x_2 & x_3 & \\ 1 & 1 & 1 & 0 & p_0 \\ - & 1 & 0 & 1 & p_1 \\ 1 & - & 0 & 0 & p_2 \\ 1 & 1 & - & 1 & p_3 \\ 1 & 1 & 1 & + & p_4 \end{array}\right|$$

Example of Execution of the Algorithm

Infer the truth table for a_3 of the matrix E shown.

$$M = \left|\begin{array}{cccc|c} x_0 & x_1 & x_2 & x_3 & \\ 1 & 1 & 1 & 0 & p_0 \\ - & 1 & 0 & 1 & p_1 \\ 1 & - & 0 & 0 & p_2 \\ 1 & 1 & - & 1 & p_3 \\ 1 & 1 & 1 & + & p_4 \end{array}\right|$$

(1) In step (1), the expression levels of a_3 differ in the row pairs (0,1), (0,3), (1,2) and (2,3). We find:

- for (0,1), $S_{01} = \{a_0, a_2\}$, containing all the other genes whose expression levels also differ in the row pairs p_0 and p_1.
- the same is done for (0,3), $S_{03} = \{a_2\}$.
- for (1,2), $S_{12} = \{a_0, a_1\}$.
- for (2,3), $S_{23} = \{a_1\}$.

Result of Step 1

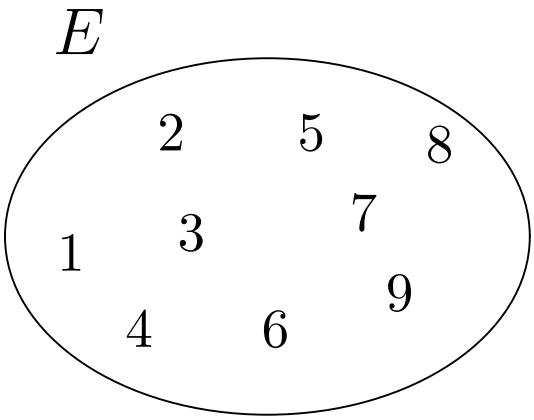
Result of Step 1: $S_{01} = \{a_0, a_2\}$, $S_{03} = \{a_2\}$, $S_{12} = \{a_0, a_1\}$, $S_{23} = \{a_1\}$.

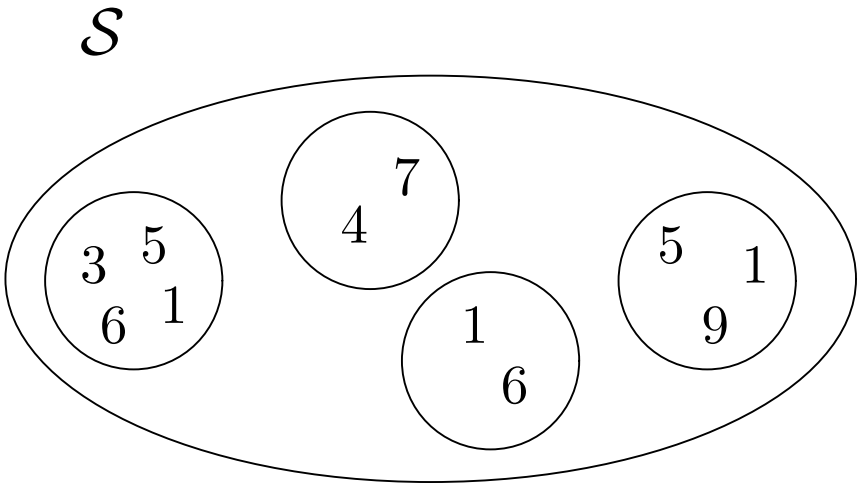
(2) In Step (2), find $S_{min} = \{a_1, a_2\}$, the smallest set such that each element in S_{min} is also present in each one of the sets S_{ij} of the previous step.

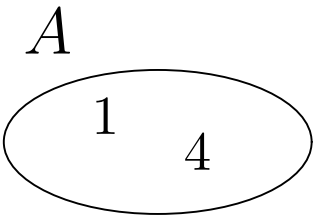
The Hitting Set Problem

- Given a finite set E, a finite collection $\mathcal{S} = \{S_1, ..., S_w\}$ of subsets of E, find a subset $A \subseteq E$ of the smallest size, such that $A \cap S_i \neq \emptyset$ for all $i = 1, ..., w$.

The Hitting Set Problem







The Hitting Set Problem

Primal-Dual Approximation Algorithm [FMCF01]

- Due to Bar-Yehuda and Even [BYE81] and was originally conceived for the minimum set cover problem.
- It is an α-approximation algorithm, where $\alpha = max_{i=1}^{w}|S_i|$.
- $\alpha = max_{i=1}^{w}|S_i| = O(n)$.

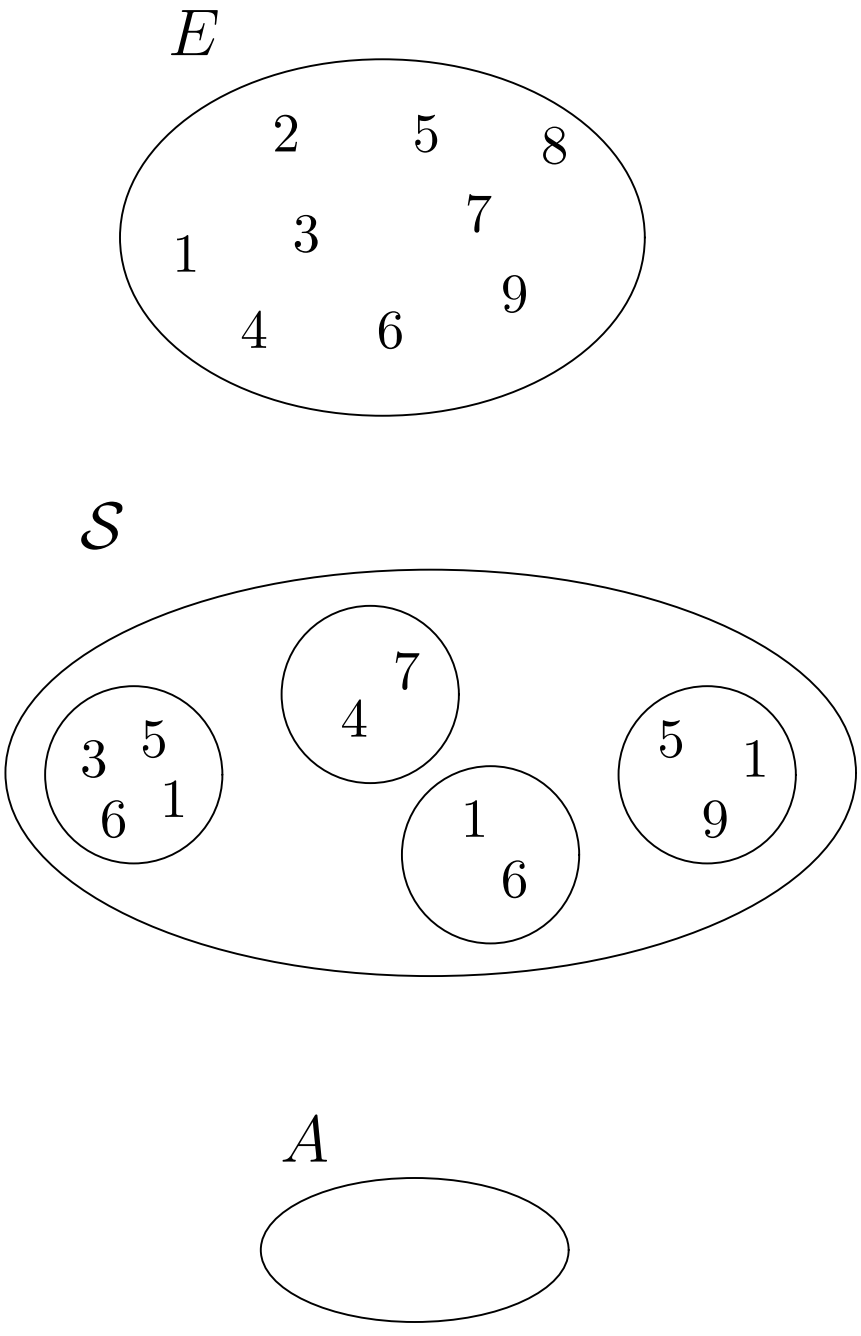
The Hitting Set Problem

Greedy Approximation Algorithm [J74]

- Strategy of constructing the set A by choosing the elements that occurs the most times in the subsets of $\mathcal{S}$.
- The approximation ratio is $\ln|\mathcal{S}| + 1$.
- $\ln|\mathcal{S}| + 1 = O(\log m^2)$

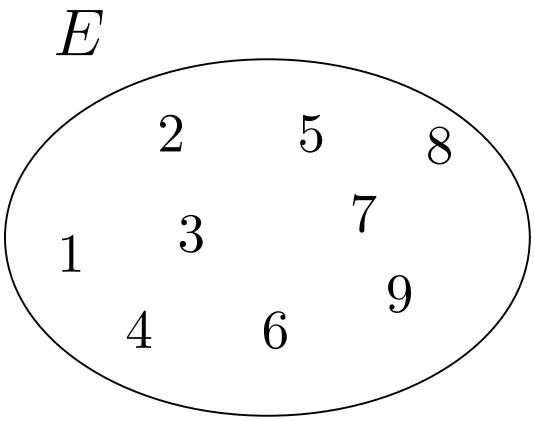
The Hitting Set Problem

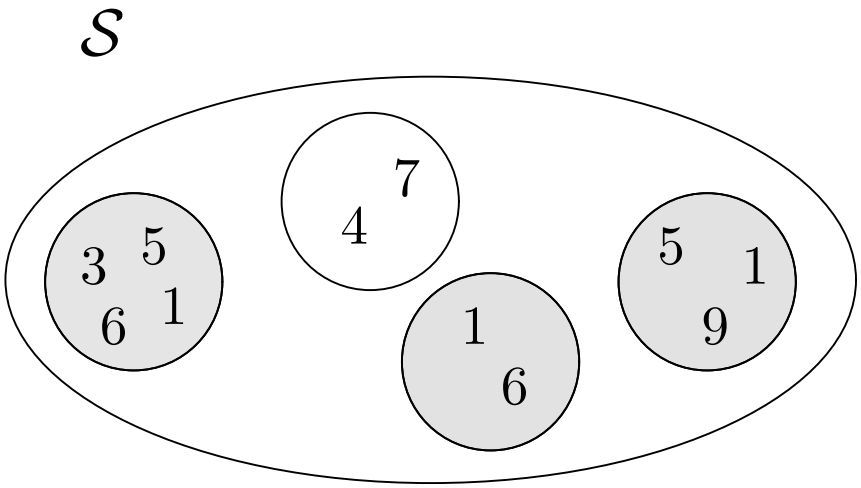
Greedy Approximation Algorithm

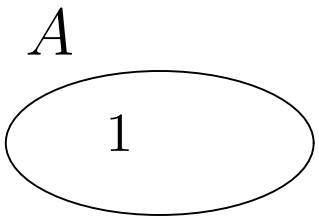


The Hitting Set Problem

Greedy Approximation Algorithm

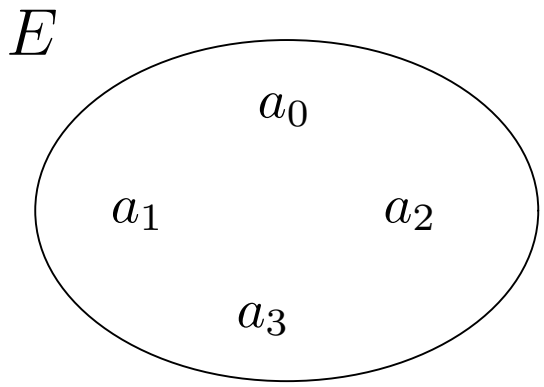


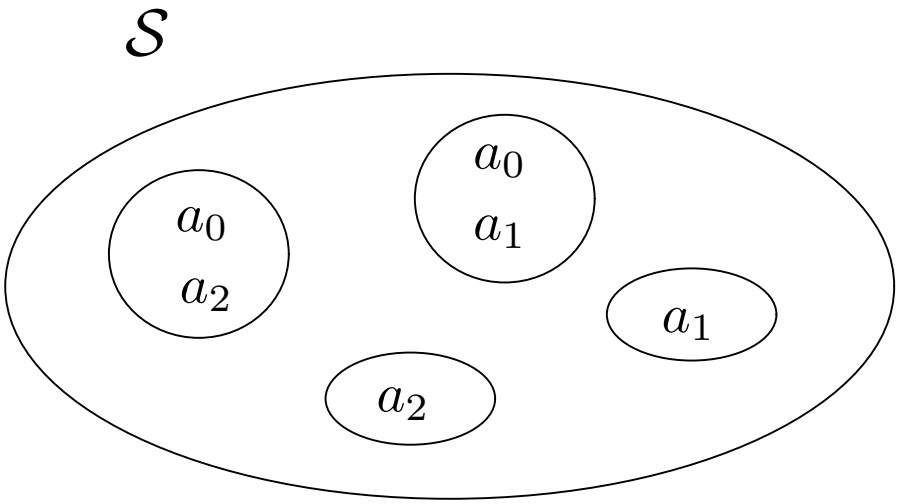


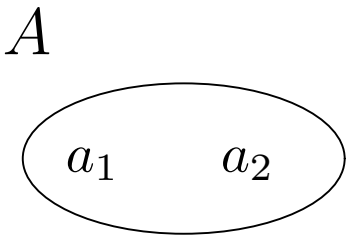


The Hitting Set Problem

Greedy Approximation Algorithm

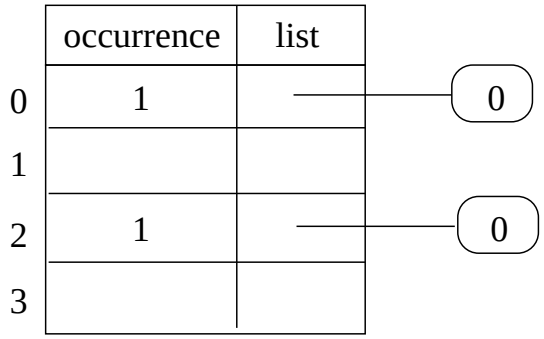




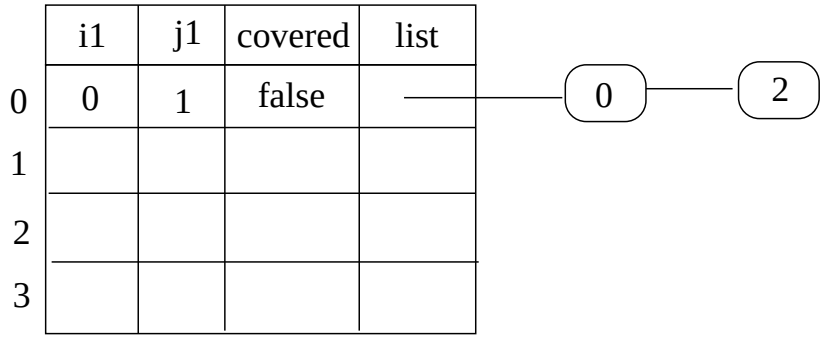


The Sequential Algorithm

gene vector



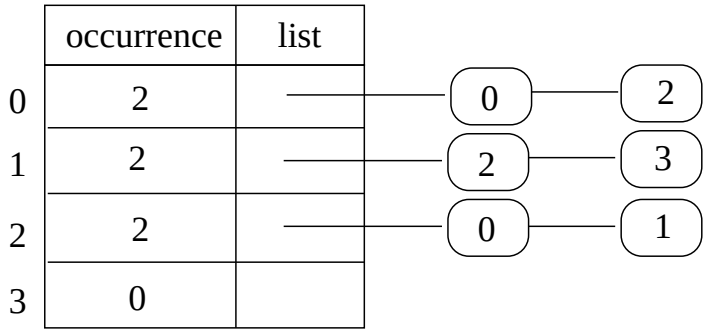
set vector



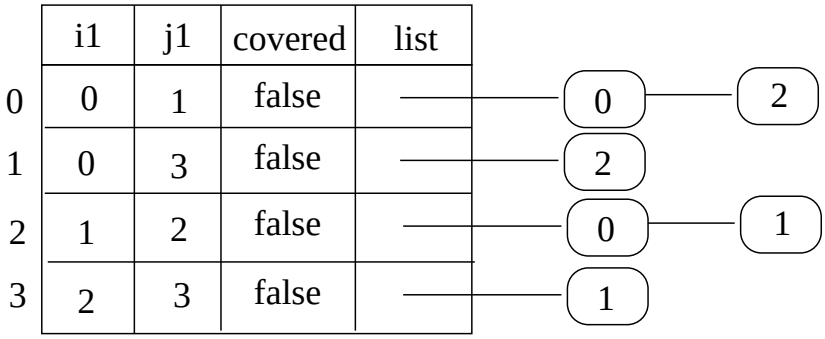
$$M = \left|\begin{array}{cccc|c} x_0 & x_1 & x_2 & x_3 & \\ 1 & 1 & 1 & 0 & p_0 \\ - & 1 & 0 & 1 & p_1 \\ 1 & - & 0 & 0 & p_2 \\ 1 & 1 & - & 1 & p_3 \\ 1 & 1 & 1 & + & p_4 \end{array}\right|$$

The Sequential Algorithm

gene vector

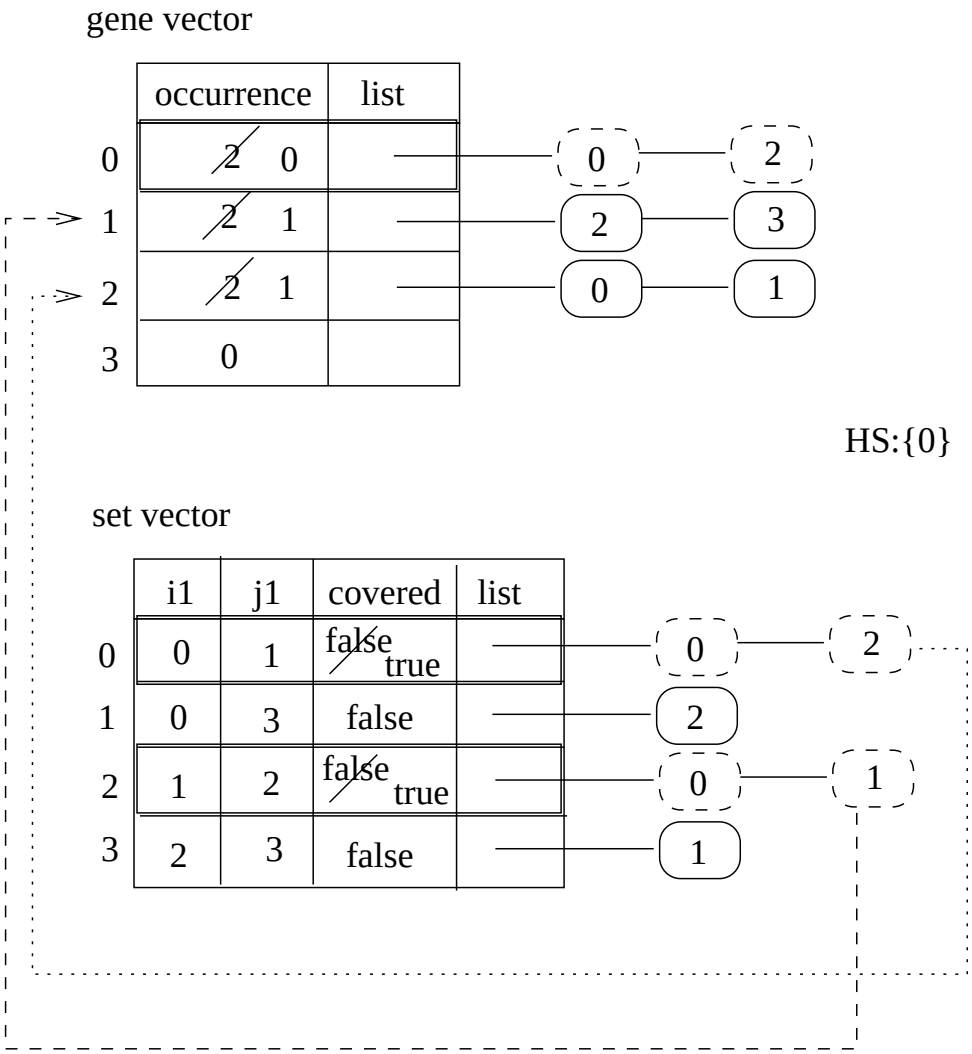


set vector



$$M = \left|\begin{array}{cccc|c} x_0 & x_1 & x_2 & x_3 & \\ 1 & 1 & 1 & 0 & p_0 \\ - & 1 & 0 & 1 & p_1 \\ 1 & - & 0 & 0 & p_2 \\ 1 & 1 & - & 1 & p_3 \\ 1 & 1 & 1 & + & p_4 \end{array}\right|$$

The Sequential Algorithm



The Sequential Algorithm

Time and Space Complexities

- To construct the data structures: $O(m^2n)$.
- Let k the size of the hitting set. We have to find k times the element with the largest number of occurrences. Therefore we have the time complexity of $O(kn)$.
- For each such element, we have to update the data structures: $O(m^2n)$ time. Since we have k elements, the total time complexity to update data structures is $O(km^2n)$.

The Sequential Algorithm

Time and Space Complexities

- The total time complexity is therefore $O(m^2n)+O(kn)+O(km^2n) = O(km^2n)$.
- The size k of the hitting set is $O(m^2)$. Therefore, the time complexity of the algorithm can be expressed as $O(m^4n)$.
- The space complexity is $O(m^2n)$.

The Parallel Algorithm

- The input matrix M is partitioned vertically to be stored in each processor.
- Example of the partitioning:

$$
\begin{array}{c|ccc:ccc:ccc}
 & a_0 & a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 \\
\hline
\mathrm{M} = & x_{0,0} & x_{0,1} & x_{0,2} & x_{0,3} & x_{0,4} & x_{0,5} & x_{0,6} & x_{0,7} & x_{0,8} \\
 & x_{1,0} & x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} & x_{1,5} & x_{1,6} & x_{1,7} & x_{1,8} \\
 & x_{2,0} & x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} & x_{2,5} & x_{2,6} & x_{2,7} & x_{2,8} \\
 & x_{3,0} & x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} & x_{3,5} & x_{3,6} & x_{3,7} & x_{3,8} \\
 & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
 & x_{m-1,0} & x_{m-1,1} & x_{m-1,2} & x_{m-1,3} & x_{m-1,4} & x_{m-1,5} & x_{m-1,6} & x_{m-1,7} & x_{m-1}
\end{array}
$$

processor 0 | processor 1 | processor 2

The Parallel Algorithm

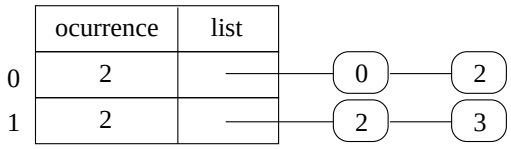
- Each processor reads a piece of the input of size $m \times \frac{n-1}{p}$.
- All the processors store a vector v, corresponding to the expression levels of the gene under study a_{n-1}.
- Each processor p_i also stores a *gene vector*, with information about genes it is responsible for. The gene vector stores information of the genes for which processor p_i is responsible. The gene vector in each processor has size $O(\frac{m^2 n}{p})$.
- Each processor also has a *set vector*, such that only elements of set S_{ij} of its responsibility will only be in the list.

The Parallel Algorithm

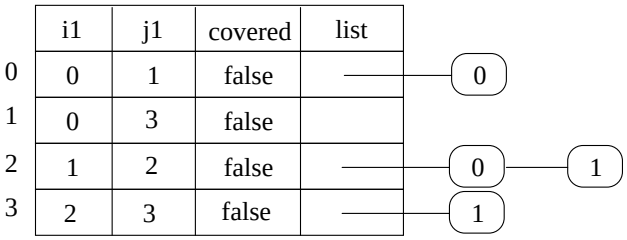
- Example:

$$E = \left|\begin{array}{ccccc|c} x_0 & x_1 & x_2 & x_3 & x_4 & \\ 1 & 1 & 1 & 0 & 0 & p_0 \\ - & 1 & 0 & 0 & 1 & p_1 \\ 1 & - & 0 & 0 & 0 & p_2 \\ 1 & 1 & - & 0 & 1 & p_3 \\ 1 & 1 & 1 & 0 & + & p_4 \end{array}\right|$$

gene vector

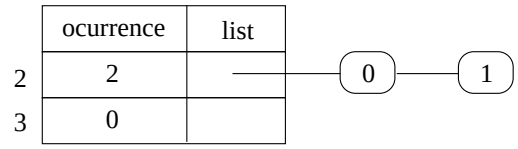


set vector

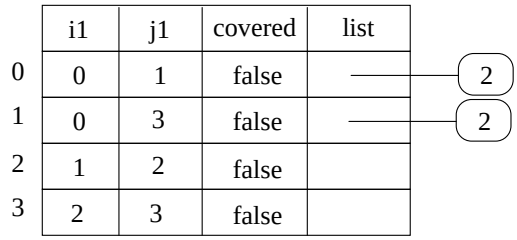


Processor 0

gene vector



set vector

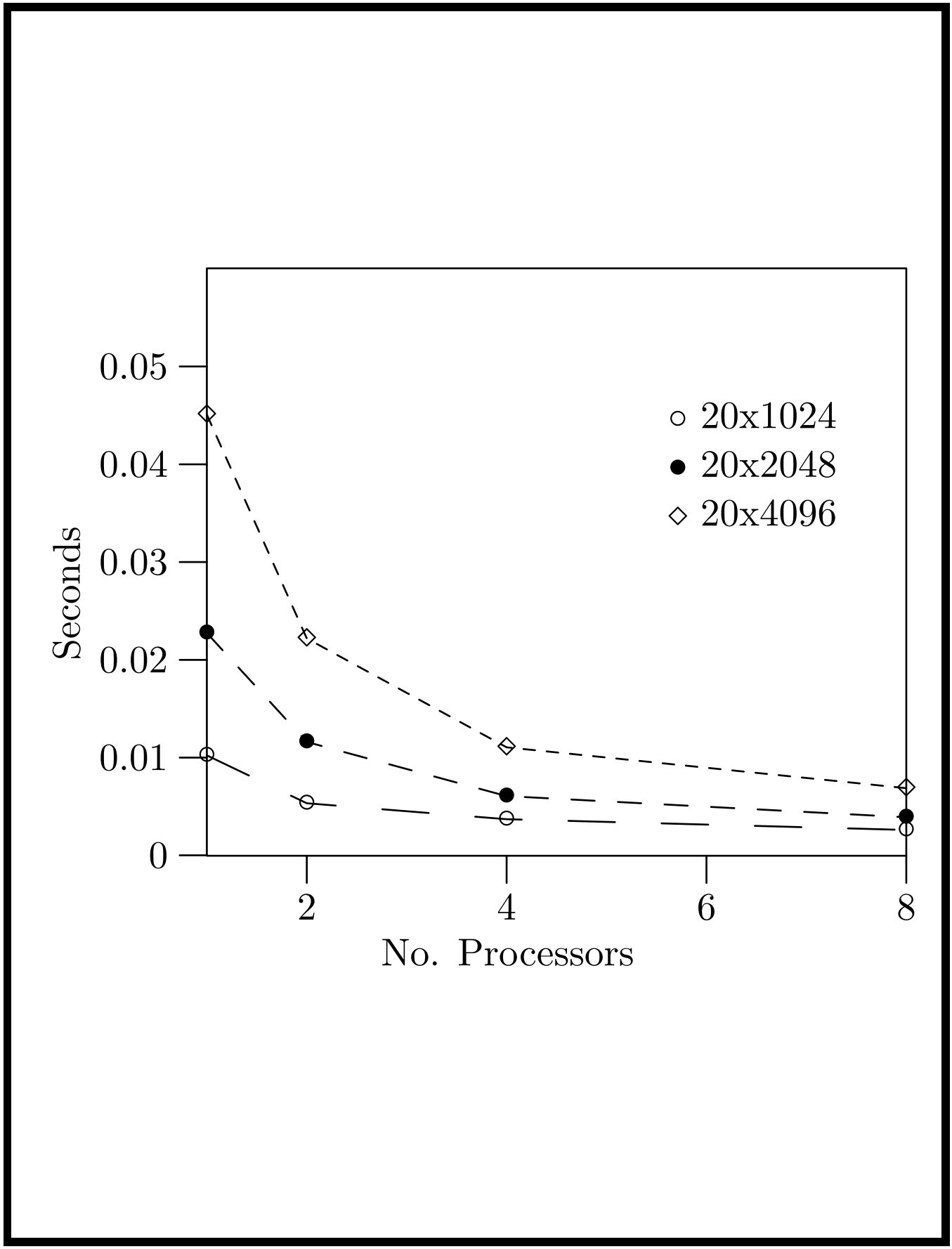


Processor 1

The Parallel Algorithm

Time and Space Complexities

- Time complexity: $O(\frac{m^4 n}{p})$.
- Requires $O(k)$ communication rounds, where k is the size of the hitting set. It can be expressed in terms of m, $O(m^2)$.
- Requires $O(\frac{m^2 n}{p})$ space.


0.05
0.04
0.03
0.02
0.01
0
Seconds
2
4
6
8
No. Processors
20x1024
20x2048
20x4096

Bibliographical References

[**BYE81**] R. Bar-Yehuda and S. Even. A linear time approximation algorithm for the weighted vertex cover problem. *Journal of Algorithms*, 2:198-203, 1981.

[**FMCF01**] C. G. Fernandes, F. K. Miyazawa, M. Cerioli, P. Feofiloff. Uma introdução sucinta a algoritmos de aproximação. *23 Colóquio Brasileiro de Matemática*, 2001.

[**ITK00**] T. E. Ideker, V. Thorsson, R. Karp. Discovery of regulatory interactions through perturbation: inference and experimental design. *Pacific Symposium on Biocomputing*, 5:302-313, 2000.

[**J74**] D. S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9:256-278, 1974.