Parallel Numerical Irreducible Decomposition

Jan Verschelde

Department of Math, Stat & CS
University of Illinois at Chicago
Chicago, IL 60607-7045, USA

email: `jan@math.uic.edu`

URL: `http://www.math.uic.edu/~jan`

Joint work with Anton Leykin (UIC).

Computational Complexity of Polynomial Factorization
American Institute of Mathematics, 15-19 May 2006.

Plan of the Talk

1. hybrid symbolic-numeric approach

   $\longrightarrow$ homotopy continuation algorithms scale well

2. numerical irreducible decomposition

   $\longrightarrow$ geometric representation $\sim$ lifting fibers

   - NC class of algorithms
   - transfer of technology from absolute factorization

3. two parallel algorithms

   $\longrightarrow$ reorganized "monodromy breakup"

Solving Polynomial Systems

assumptions

Consider $f(\mathbf{x}) = \mathbf{0}$, N equations in n variables ...

- degrees of polynomials in f are low
- coefficients are **approximate** numbers in $\mathbb{C}$
- floating-point **approximations** to solutions
- $N = n$: square systems, expect $\dim f^{-1}(\mathbf{0}) = 0$

Numerical Homotopy Algorithms

1. deform f into a generic system $g(\mathbf{x}) = \mathbf{0}$
2. define a homotopy, typically like

$$h(\mathbf{x}, t) = \gamma(1 - t)g(\mathbf{x}) + tf(\mathbf{x}) = \mathbf{0},$$

   $\gamma \in \mathbb{C}$ random $\Rightarrow$ regular solution paths for all t: $0 \leq t < 1$

3. apply numerical path tracking algorithms

M. Shub and S. Smale. **Complexity of Bézout's Theorem I: Geometric Aspects**. *J. Amer. Math. Soc.* 6(2):459–501,1993.

Scalability for Parallel Computers

Work of Layne T. Watson & collaborators (1989, 1991, 1993)
$\rightarrow$ one job = track one path.

Dynamic load balancing needed ...

... when not all paths need same computational cost.

Joint work with Yusong Wang (HPSEC'04).

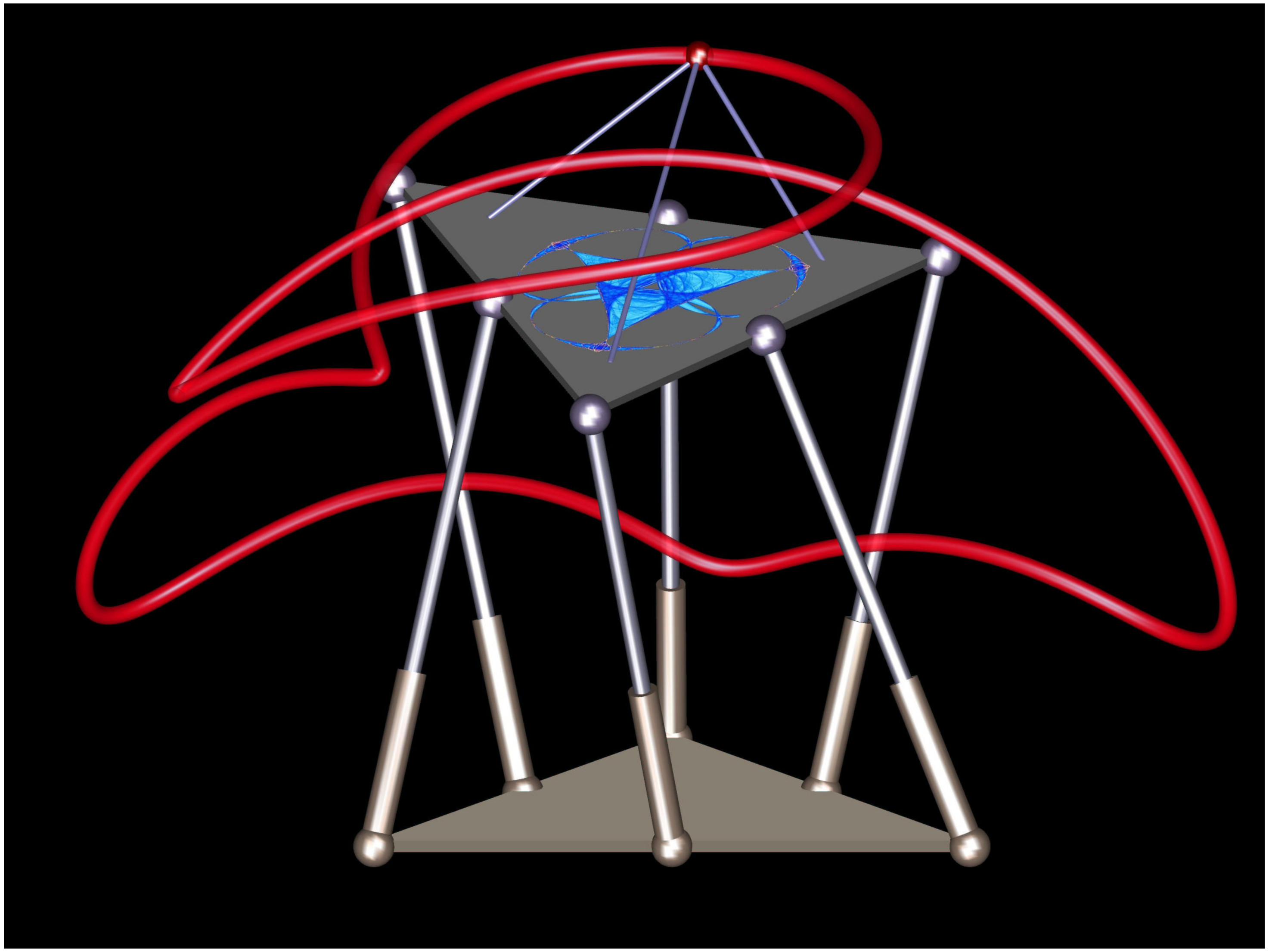
On our cluster with 14 cpu's at 2.4Ghz ...

... we can run about 100,000 paths "overnight".

Equations for a Griffis-Duffy Platform

```
  0.427419669081*e0 + 0.321110693270*e1 + 0.343633073697*e2 + 0.474256143563*e3 + 0.558458718976;

  0.4754797951714257*e0^2 + 1.6317349621954614*e0*e3 + 0.9465211050062624*e1^2 - 0.9376441343330845*e2^2
- 0.4666028244982478*e3^2 + 2.55*g0*e1 + 0.2598076211353316*g0*e2 - 2.55*g1*e0 - 1.4722431864335457*g1*e3
- 0.2598076211353316*g2*e0 + 0.45*g2*e3 + 1.4722431864335457*g3*e1 - 0.45*g3*e2;

- 0.2385669606907614*e0^2 + 3.2634699243909228*e0*e3 + 1.6455982786485855*e1^2 - 3.2634699243909228*e1*e2
- 0.2385669606907614*e2^2 + 1.6455982786485855*e3^2 + 3*g0*e1 + 0.5196152422706632*g0*e2 - 3*g1*e0
- 2.9444863728670914*g1*e3 - 0.5196152422706632*g2*e0 + 3*g2*e3 + 2.9444863728670914*g3*e1 - 3*g3*e2;

- 0.3961611384685069*e0^2 + 4.8952048865863842*e0*e3 - 0.3961611384685069*e1^2 - 4.8952048865863842*e1*e2
+ 2.4300867205405135*e2^2 + 2.4300867205405135*e3^2 + 1.95*g0*e1 - 1.8186533479473212*g0*e2 - 1.95*g1*e0
- 1.8186533479473212*g1*e3 + 1.8186533479473212*g2*e0 + 4.05*g2*e3 + 1.8186533479473212*g3*e1 - 4.05*g3*e2;

  1.2028336489822089*e0^2 + 3.2634699243909228*e0*e3 + 0.2607510293125354*e1^2 + 4.0290815079912293*e2^2
+ 3.0869988883215558*e3^2 + 1.95*g0*e1 - 2.3382685902179843*g0*e2 - 1.95*g1*e0 + 1.1258330249197702*g1*e3
+ 2.3382685902179843*g2*e0 + 4.05*g2*e3 - 1.1258330249197702*g3*e1 - 4.05*g3*e2;

  0.8849480315885287*e0^2 + 1.3559893414233654*e1^2 + 1.6317349621954614*e1*e2 + 2.2980719610930389*e2^2
+ 2.7691132709278756*e3^2 + 0.45*g0*e1 - 0.2598076211353316*g0*e2 - 0.45*g1*e0 + 1.4722431864335457*g1*e3
+ 0.2598076211353316*g2*e0 + 2.55*g2*e3 - 1.4722431864335457*g3*e1 - 2.55*g3*e2;

g0*e0+g1*e1+g2*e2+g3*e3;
g0^2+g1^2+g2^2+g3^2-e0^2-e1^2-e2^2-e3^2;
```

Irreducible Factor = Assembly of Platform



Adjacent minors of a general 2-by-$(n+1)$ matrix

$$n = 3 : \quad \begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \end{bmatrix} \qquad f(\mathbf{x}) = \begin{cases} x_{11}x_{22} - x_{21}x_{12} = 0 \\ x_{12}x_{23} - x_{22}x_{13} = 0 \\ x_{13}x_{24} - x_{23}x_{14} = 0 \end{cases}$$

P. Diaconis, D. Eisenbud, and B. Sturmfels. **Lattice walks and primary decomposition.** In *Mathematical Essays in Honor of Gian-Carlo Rota*, ed. B.E. Sagan and R.P. Stanley, pages 173–193, Birkhäuser, 1998.

S. Hoşten and J.Shapiro. **Primary decomposition of lattice basis ideals.** *J. Symbolic Computation* 29(4&5): 625–639.

Challenge in Computational Algebraic Geometry

O. Holtz and B. Sturmfels. **Hyperdeterminantal relations among symmetric principal minors.** arXiv:math.RA/0604374 v1.

The principal minors of a real symmetric 4-by-4 matrix form a vector of length 2^4 and satisfy the hyperdeterminantal relations of format $2 \times 2 \times 2$ (derived from Schur's identity).

$\rightarrow$ 8 equations in 16 unknowns

product of the degrees is 122,880 ...

Absolute Factorization

"*Given is a polynomial $f(x,y) \in \mathbb{Q}[x,y]$ and $\epsilon \in \mathbb{Q}$. Decide in polynomial time in the degree and coefficient size if there is a factorizable $\hat{f}(x,y) \in \mathbb{C}[x,y]$ with $||f - \hat{f}|| \leq \epsilon$, for a reasonable coefficient vector norm $||.||$.*"

Erich Kaltofen (JSC 29, 2000) (originally in Kaltofen, 1992)

Recent Work: new symbolic-numeric algorithms by

- T. Sasaki, T. Saito, T. Hilano (1992); T. Sasaki (2001)
- A. Galligo, D. Rupprecht (2001); G. Chèze, A. Galligo (2003)
- R.M. Corless, M.W. Giesbrecht, M. van Hoeij, I.S. Kotsireas, S.M. Watt (2001)
- A.J. Sommese, J. Verschelde, C.W. Wampler (2001, 2002, 2004)
- S. Gao, E. Kaltofen, J. May, Z. Yang, L. Zhi (2004)

complexity class NC

NC is the class of functions computable by logspace-uniform boolean circuits of polynomial size and polylogarithmic depth.

E. Kaltofen. **Fast parallel absolute irreducibility testing**. JSC 1985.

→ **NC algorithm for absolute factorization**

C. Bajaj, J. Canny, T. Garrity, J. Warren. **Factoring rational polynomials over the complex numbers.** ISSAC 1989, SICOMP 1993.

→ **this is a topological problem, not an algebraic one**

Representations: ϵ-gcd and δ-gcd

following Victor Pan 2001

Consider two univariate polynomials $p, q \in \mathbb{C}[x]$

- with approximate coefficients;
- with approximate roots.

GCD $h(x)$ satisfies $h(x) = k(x)p(x) + l(x)q(x)$

δ-gcd: match roots of p and q
find $k(x)$ and $l(x)$ via interpolation

ϵ-gcd: rank revealing on Sylvester matrix
Zhonggang Zeng (ISSAC 2003)

δ-gcd $\approx$ ϵ-gcd modulo conditioning, otherwise δ-gcd $\neq$ ϵ-gcd.

Geometric View on an Algebraic Set V

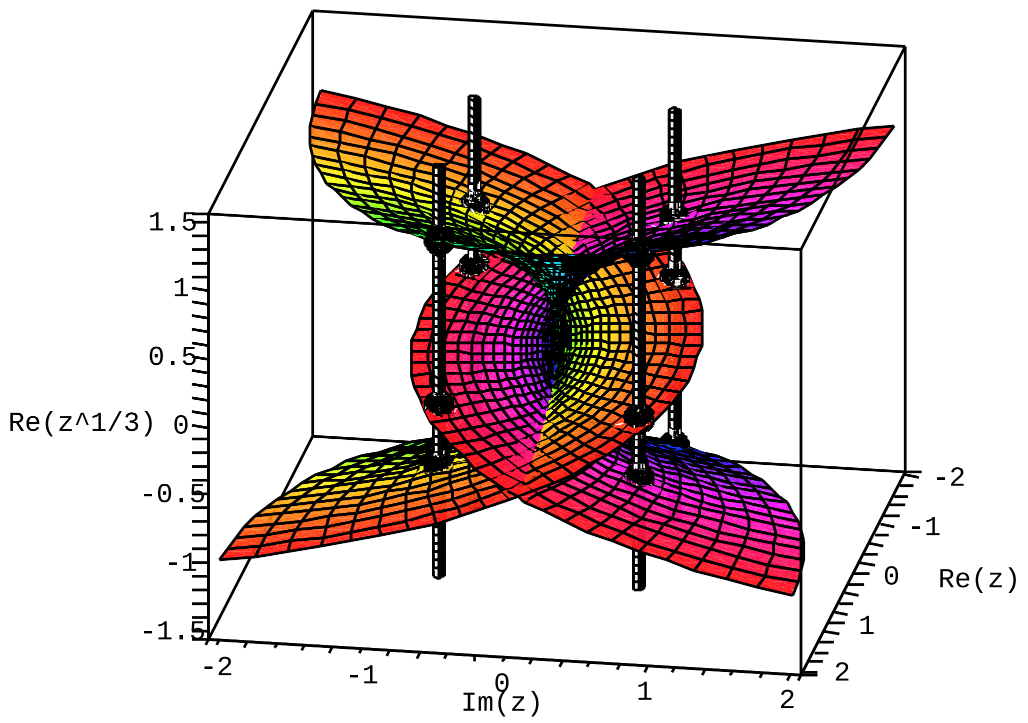
When we cut V with $\dim(V)$ many generic hyperplanes L,

we find $\deg(V)$ many regular isolated solutions in $V \cap L$.

Geometric Representations of V:

- exact: lifting fibers
  (M. Giusti, G. Lecerf, and B. Salvy, 2001)
- numerical: witness set
  (A.J. Sommese, J. Verschelde, C.W. Wampler, 2003)

The Riemann Surface of $z - w^3 = 0$:



Loop around the singular point (0,0) permutes the points.

Generating Loops by Homotopies

W_L represents a k-dimensional solution set of $f(\mathbf{x}) = \mathbf{0}$, cut out by k random hyperplanes L. For k other hyperplanes K, we move W_L to W_K, using the **homotopy** $\boldsymbol{h}_{L,K,\boldsymbol{\alpha}}(\mathbf{x}, t) = 0$, <u>from $t = 0$ to 1:</u>

$$\boldsymbol{h}_{L,K,\boldsymbol{\alpha}}(\mathbf{x}, t) = \left(\begin{array}{c} f(\mathbf{x}) \\ \boldsymbol{\alpha}(1-t)L(\mathbf{x}) + tK(\mathbf{x}) \end{array} \right) = \mathbf{0}, \quad \boldsymbol{\alpha} \in \mathbb{C}.$$

The constant $\boldsymbol{\alpha}$ is chosen at random, to avoid singularities, as $t < 1$.

To turn back we generate another random constant $\boldsymbol{\beta}$, and use

$$\boldsymbol{h}_{K,L,\boldsymbol{\beta}}(\mathbf{x}, t) = \left(\begin{array}{c} f(\mathbf{x}) \\ \boldsymbol{\beta}(1-t)K(\mathbf{x}) + tL(\mathbf{x}) \end{array} \right) = \mathbf{0}, \quad \boldsymbol{\beta} \in \mathbb{C}.$$

A permutation of points in W_L occurs only among points on the same irreducible component.

Linear Traces as Stop Criterium

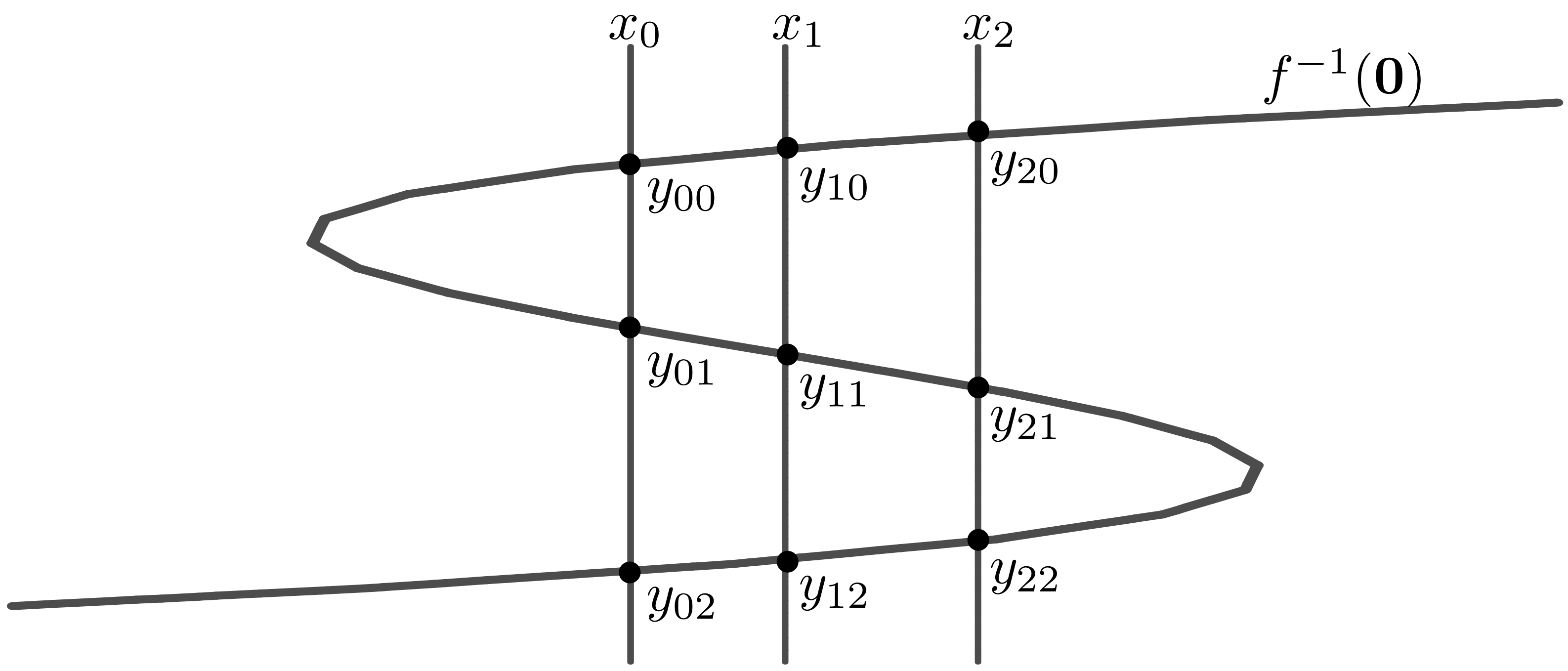
$$\text{Consider} \quad f(x,y(x)) = (y-y_1(x))(y-y_2(x))(y-y_3(x))$$
$$= y^3 - \boldsymbol{t_1(x)}y^2 + t_2(x)y - t_3(x)$$

We are interested in **the linear trace**: $\boldsymbol{t_1(x) = c_1 x + c_0}$.
Sample the cubic at $x = x_0$ and $x = x_1$. The samples are
$\{(x_0, y_{00}), (x_0, y_{01}), (x_0, y_{02})\}$ and $\{(x_1, y_{10}), (x_1, y_{11}), (x_1, y_{12})\}$.

$$\text{Solve} \quad \left\{ \begin{array}{l} y_{00} + y_{01} + y_{02} = c_1 x_0 + c_0 \\ y_{10} + y_{11} + y_{12} = c_1 x_1 + c_0 \end{array} \right. \quad \text{to find} \quad c_0, c_1.$$

With t_1 we can predict the sum of the y's for a fixed choice of x.
For example, samples at $x = x_2$ are $\{(x_2, y_{20}), (x_2, y_{21}), (x_2, y_{22})\}$.
Then, $t_1(x_2) = c_1 x_2 + c_0 = y_{20} + y_{21} + y_{22}$.
If $\neq$, then samples come from irreducible curve of degree > 3.

Linear Traces – an example



Use $\{(x_0, y_{00}), (x_0, y_{01}), (x_0, y_{02})\}$ and $\{(x_1, y_{10}), (x_1, y_{11}), (x_1, y_{12})\}$ to find the linear trace $t_1(x) = c_0 + c_1 x$.

At $\{(x_2, y_{20}), (x_2, y_{21}), (x_2, y_{22})\}$: $c_0 + c_1 x_2 = y_{20} + y_{21} + y_{22}$?

Griffis-Duffy Platforms: Factorization

Case A: One irreducible component of degree 28 (general case).

Case B: Five irreducible components of degrees 6, 6, 6, 6, and 4.

user cpu on 800Mhz	Case A	Case B
witness points	1m 12s 480ms	
monodromy breakup	**33s 430ms**	**27s 630ms**
Newton interpolation	1h 19m 13s 110ms	2m 34s 50ms

32 decimal places used to interpolate polynomial of degree 28

linear trace	4s 750ms	4s 320ms

Linear traces replace Newton interpolation:

$\Rightarrow$ **time to factor independent of geometry!**

Computational results on adjacent minors

n	d	$\#f$	witness set	#loops	factorization
3	8	3	1.4 s	9	6.8 s
4	16	5	4.5 s	3	9.4 s
5	32	8	23.9 s	4	41.6 s
6	64	13	56.4 s	2	1 m 17.0 s
7	128	21	3 m 39.5 s	4	6 m 42.0 s
8	256	34	8 m 22.6 s	5	16 m 54.7 s
9	512	55	25 m 19.2 s	7	1 h 48 m 52.9 s
10	1024	89	1 h 9 m 27.0 s	5	2 h 9 m 5.1 s

on 1 Ghz PowerBook G4 Mac OS X 10.3.4 with gcc 3.3

Monodromy Breakup certified by Linear Traces

```
Input: $W_L$, $d$, $N$                                              witness set, degree, #loops
Output: $\mathcal{P}$                                               partitioned witness set

0. initialize $\mathcal{P}$ with $d$ singletons;                    done by manager
1. generate two slices $L'$ and $L''$ parallel to $L$;              broadcast data to nodes
2. track $d$ paths for witness set with $L'$;                       executed in parallel
3. track $d$ paths for witness set with $L''$;                      executed in parallel
4. for $k$ from 1 to $N$ do
    4.1 generate new slices $K$ and a random $\alpha$;              broadcast $K$ and $\alpha$
    4.2 track $d$ paths defined by $h_{L,K,\alpha}(\mathbf{x}, t) = \mathbf{0}$;   executed in parallel
    4.3 generate a random $\beta$;                                  broadcast $\beta$ to nodes
    4.4 track $d$ paths defined by $h_{K,L,\beta}(\mathbf{x}, t) = \mathbf{0}$;    executed in parallel
    4.5 compute the permutation and update $\mathcal{P}$;           done by manager
    4.6 exit when linear trace test certifies $\mathcal{P}$.
```

A Benchmark Example: cyclic 8-roots

The system

$$f(\mathbf{x}) = \left\{ \begin{array}{l} f_i = \displaystyle\sum_{j=0}^{7} \prod_{k=1}^{i} x_{(k+j) \bmod 8} = 0, \quad i = 1, 2, \ldots, 7 \\ f_8 = x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7 - 1 = 0 \end{array} \right.$$

has 1152 isolated solutions and a solution curve of degree 144, which breaks up into 16 irreducible factors.

There are 8 factors of degree 16, and 8 quadratic factors.

Our equipment consists of one workstation with two dual 2.4Ghz processors, running Linux, and serving two Rocketcalc clusters: one with four and an other with eight 2.4Ghz processors. So we have a total of **14 processors: a manager and 13 workers.**

Computational Results (HPSEC'05)

- Fluctuations in **work loads** and influence of **number of loops** needed:

#L	4	5	6	7	7	7	7	7	8	9
min	6.0	7.8	9.2	10.1	10.3	10.9	10.9	10.7	11.8	12.3
max	9.9	11.5	12.8	15.4	15.1	14.7	14.1	14.5	16.3	16.9
total	11.7	14.9	16.9	19.2	19.3	19.5	19.7	20.3	21.9	23.4

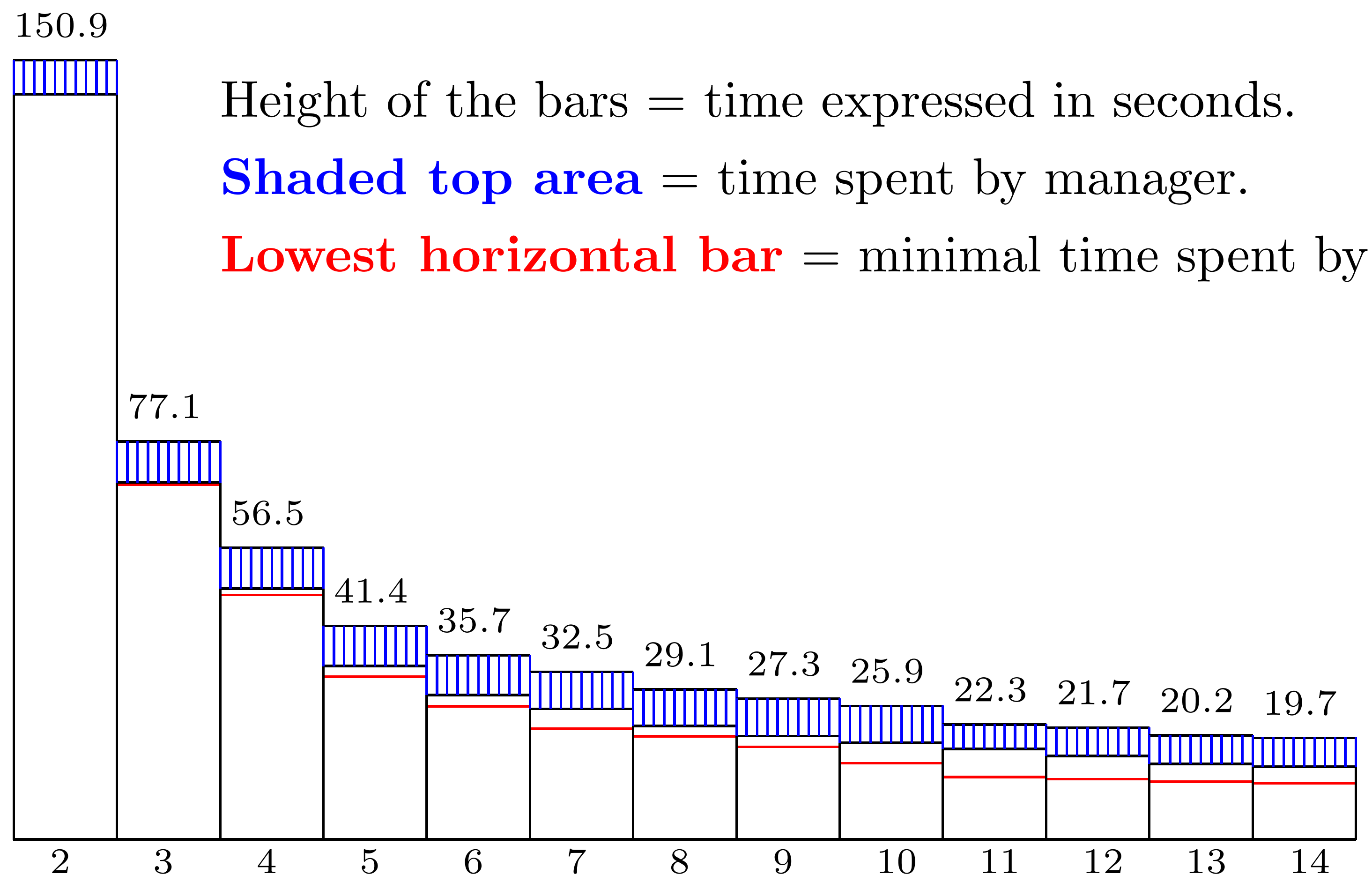
Results of 10 runs on 14 processors. **#L = number of loops**, **min** and **max** are the minimal and maximal time (in seconds) spent by the worker nodes.

- **Speedup**:

NP	2	3	4	5	6	7	8	9	10	11	12	13	14
min	—	68.7	47.4	31.5	25.8	21.5	20.0	18.0	14.8	12.1	11.7	11.2	10.9
max	144.3	69.2	48.6	33.6	28.0	25.3	22.0	20.1	18.8	17.6	16.2	14.7	14.1
total	150.9	77.1	56.5	41.4	35.7	32.5	29.1	27.3	25.9	22.3	21.7	20.2	19.7

Execution times for number of processors NP, from 2 to 14, **using 7 loops**.

Performance of a First Parallel Implementation



Height of the bars = time expressed in seconds.

Shaded top area = time spent by manager.

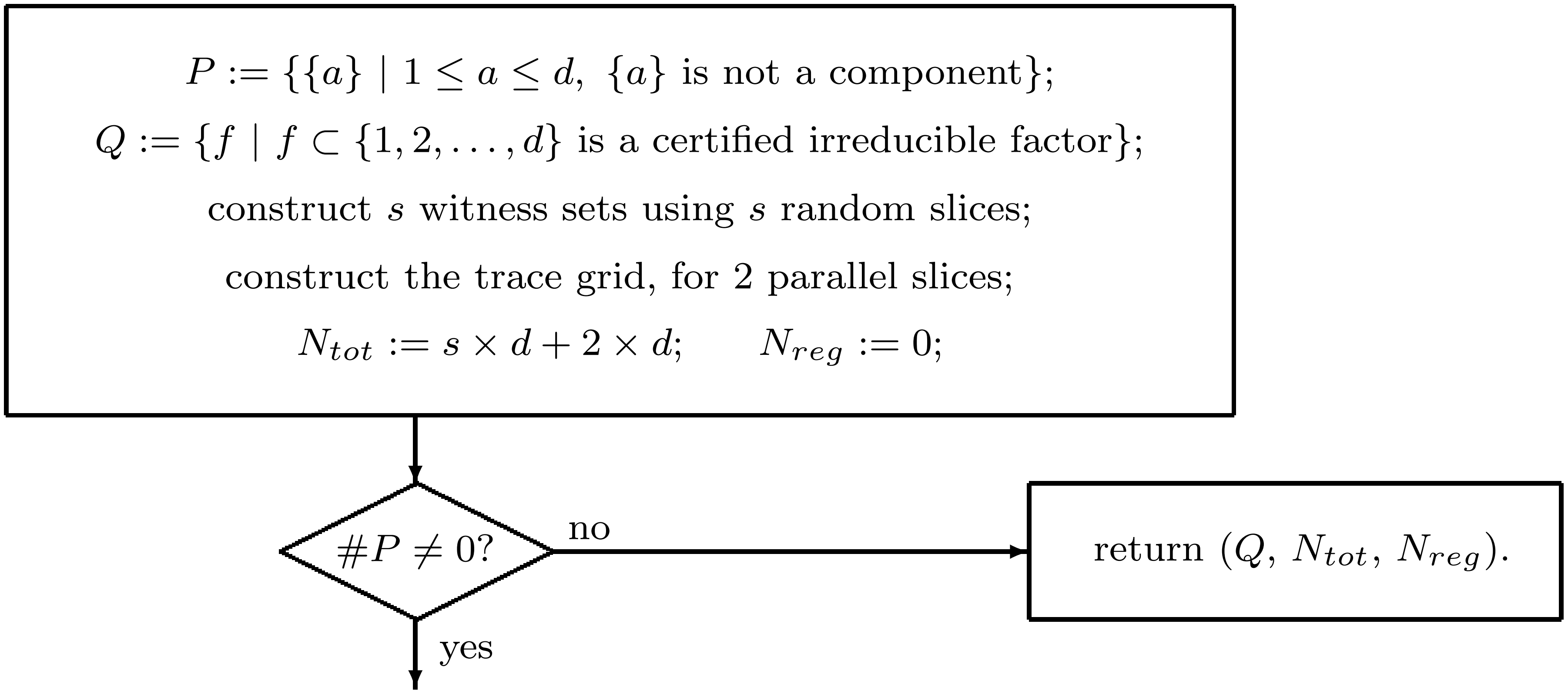
Lowest horizontal bar = minimal time spent by worker.

Number of Processors

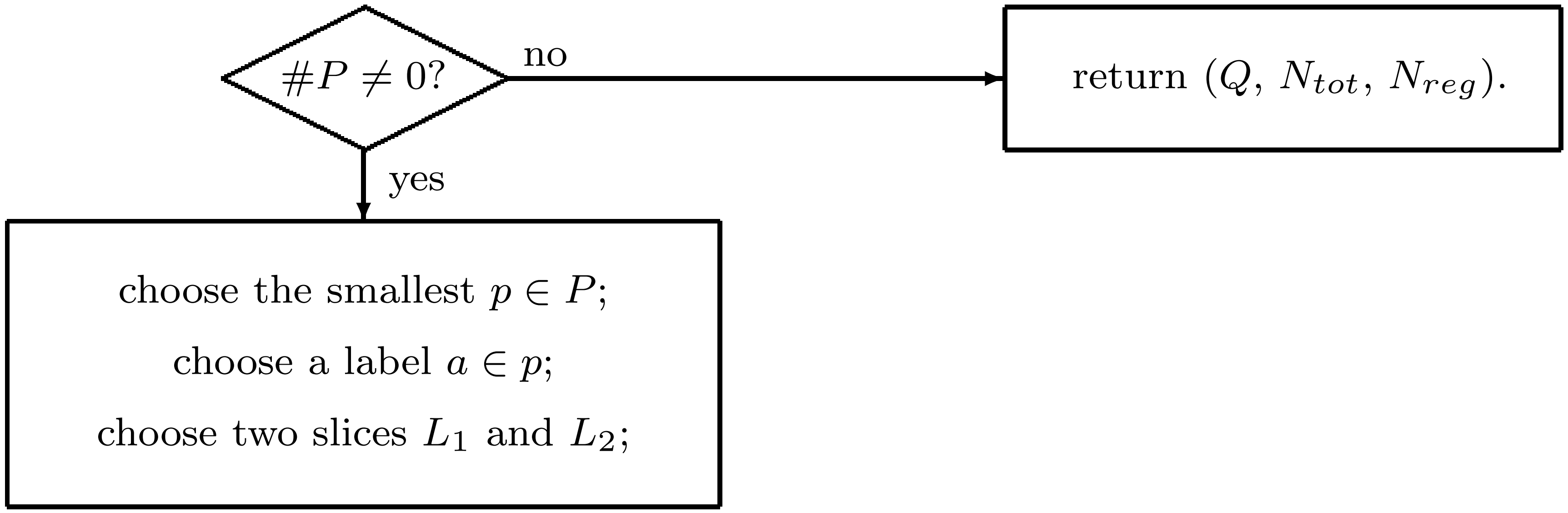
The New Algorithm – Serial Version

```
$P := \{\{a\} \mid 1 \leq a \leq d,\ \{a\}$ is not a component$\}$;
$Q := \{f \mid f \subset \{1, 2, \ldots, d\}$ is a certified irreducible factor$\}$;
construct $s$ witness sets using $s$ random slices;
construct the trace grid, for 2 parallel slices;
$N_{tot} := s \times d + 2 \times d$;      $N_{reg} := 0$;
```

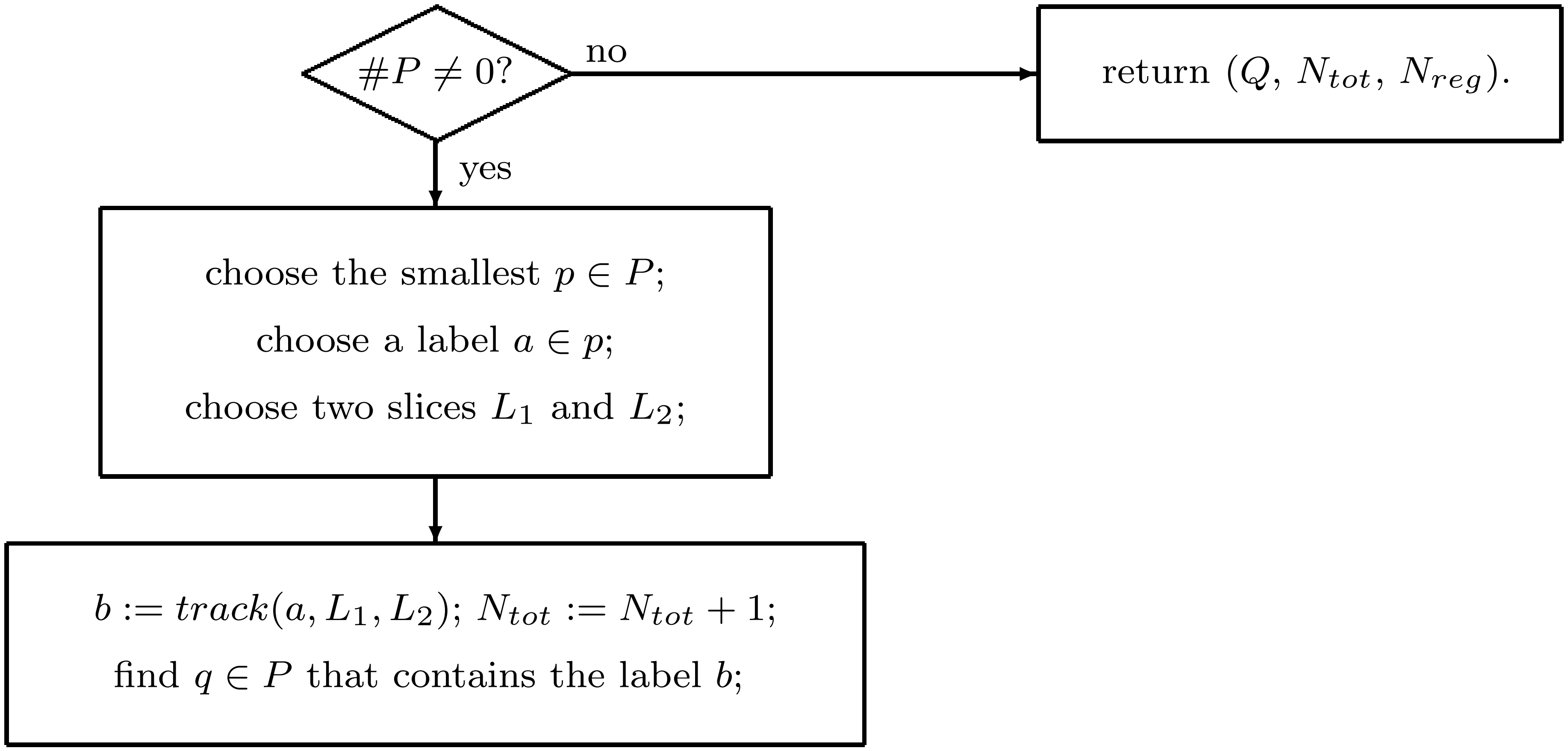
The initialization requires $(s+2)d$ paths


$P := \{\{a\} \mid 1 \leq a \leq d,\ \{a\}$ is not a component$\}$;
$Q := \{f \mid f \subset \{1, 2, \ldots, d\}$ is a certified irreducible factor$\}$;
construct s witness sets using s random slices;
construct the trace grid, for 2 parallel slices;
$N_{tot} := s \times d + 2 \times d$; $N_{reg} := 0$;
$\#P \neq 0$?
no
return $(Q,\ N_{tot},\ N_{reg})$.
yes

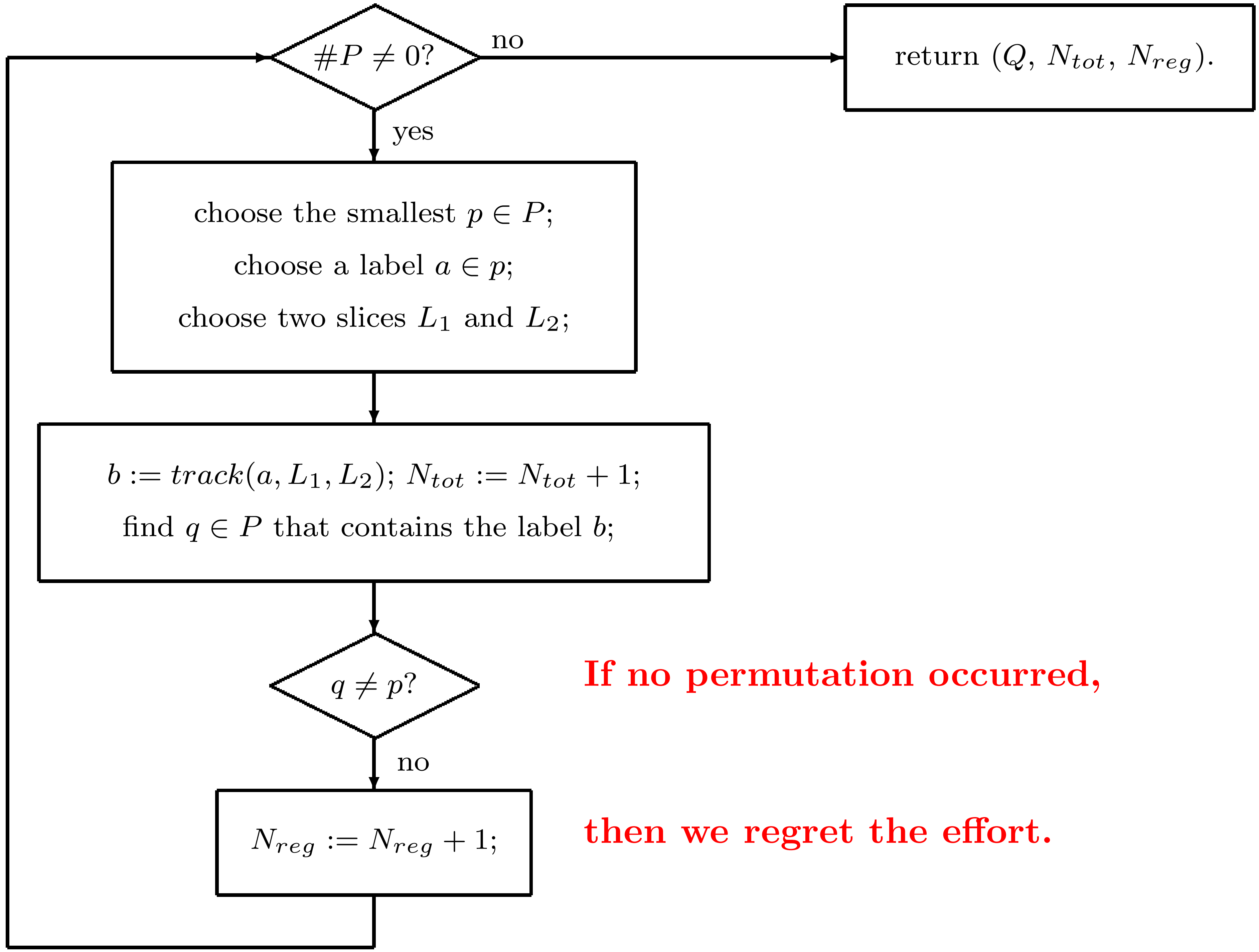
On return is a certified factorization,
and path tracking statistics.


$\#P \neq 0?$
no
return $(Q,\ N_{tot},\ N_{reg})$.
yes
choose the smallest $p \in P$;
choose a label $a \in p$;
choose two slices L_1 and L_2;

Use path tracking statistics to discriminate against nonproductive slices.

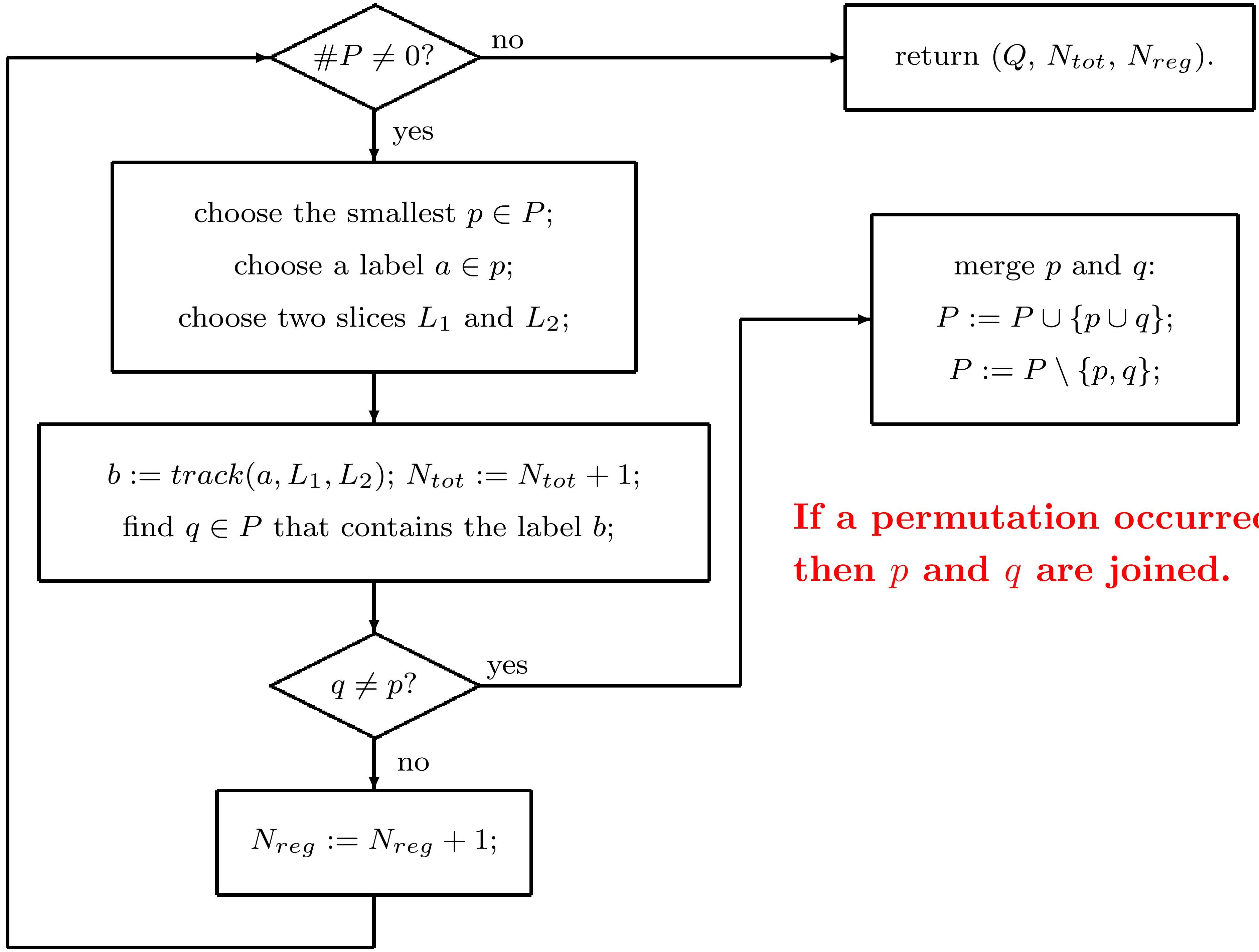

$\#P \neq 0?$
no
return $(Q,\ N_{tot},\ N_{reg})$.
yes
choose the smallest $p \in P$;
choose a label $a \in p$;
choose two slices L_1 and L_2;
$b := track(a, L_1, L_2)$; $N_{tot} := N_{tot} + 1$;
find $q \in P$ that contains the label b;

Track a path to close a loop.

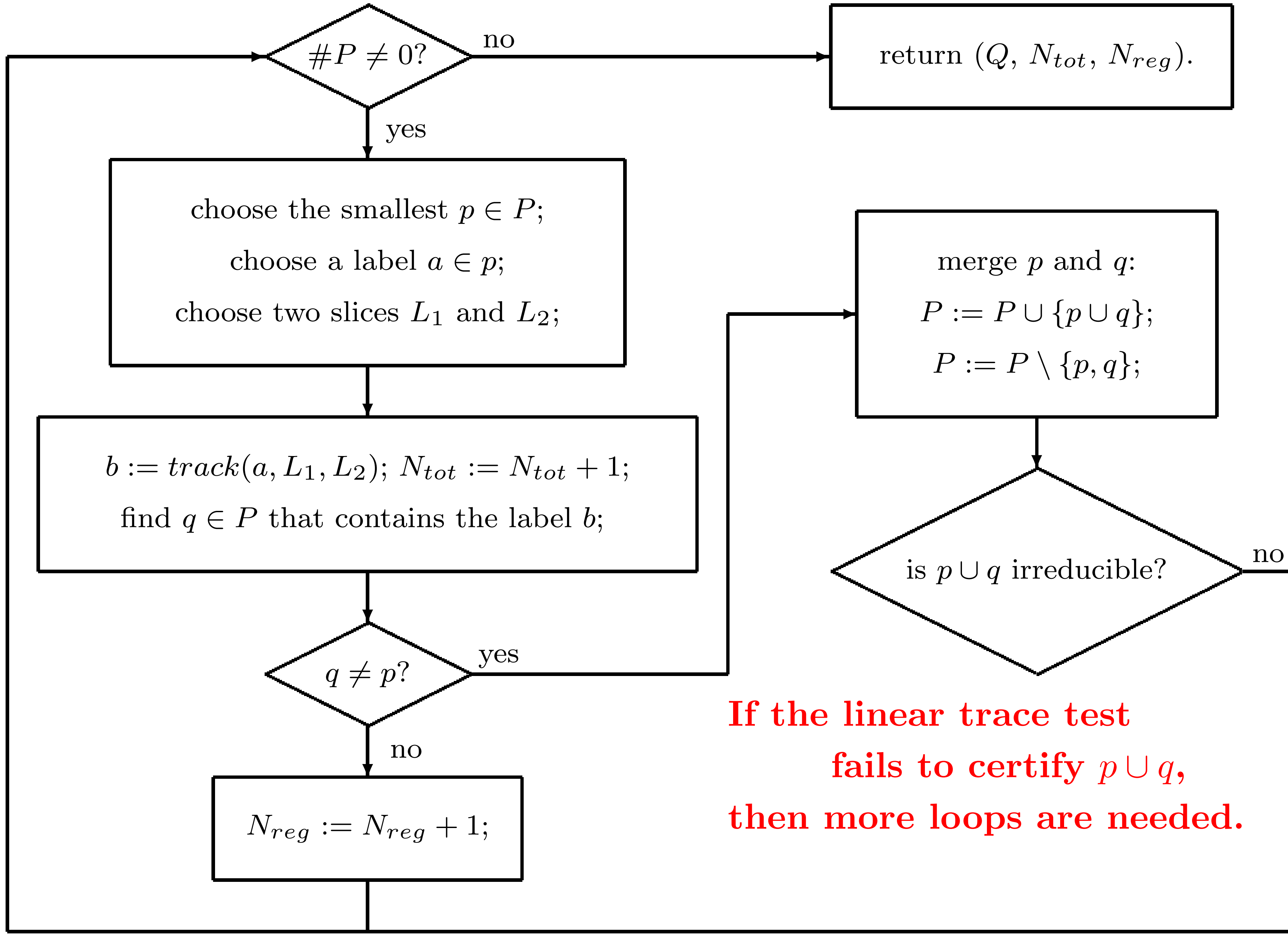

$\#P \neq 0?$
no
return $(Q,\ N_{tot},\ N_{reg})$.
yes
choose the smallest $p \in P$;
choose a label $a \in p$;
choose two slices L_1 and L_2;
$b := track(a, L_1, L_2)$; $N_{tot} := N_{tot} + 1$;
find $q \in P$ that contains the label b;
$q \neq p?$
no
$N_{reg} := N_{reg} + 1$;

If no permutation occurred,

then we regret the effort.

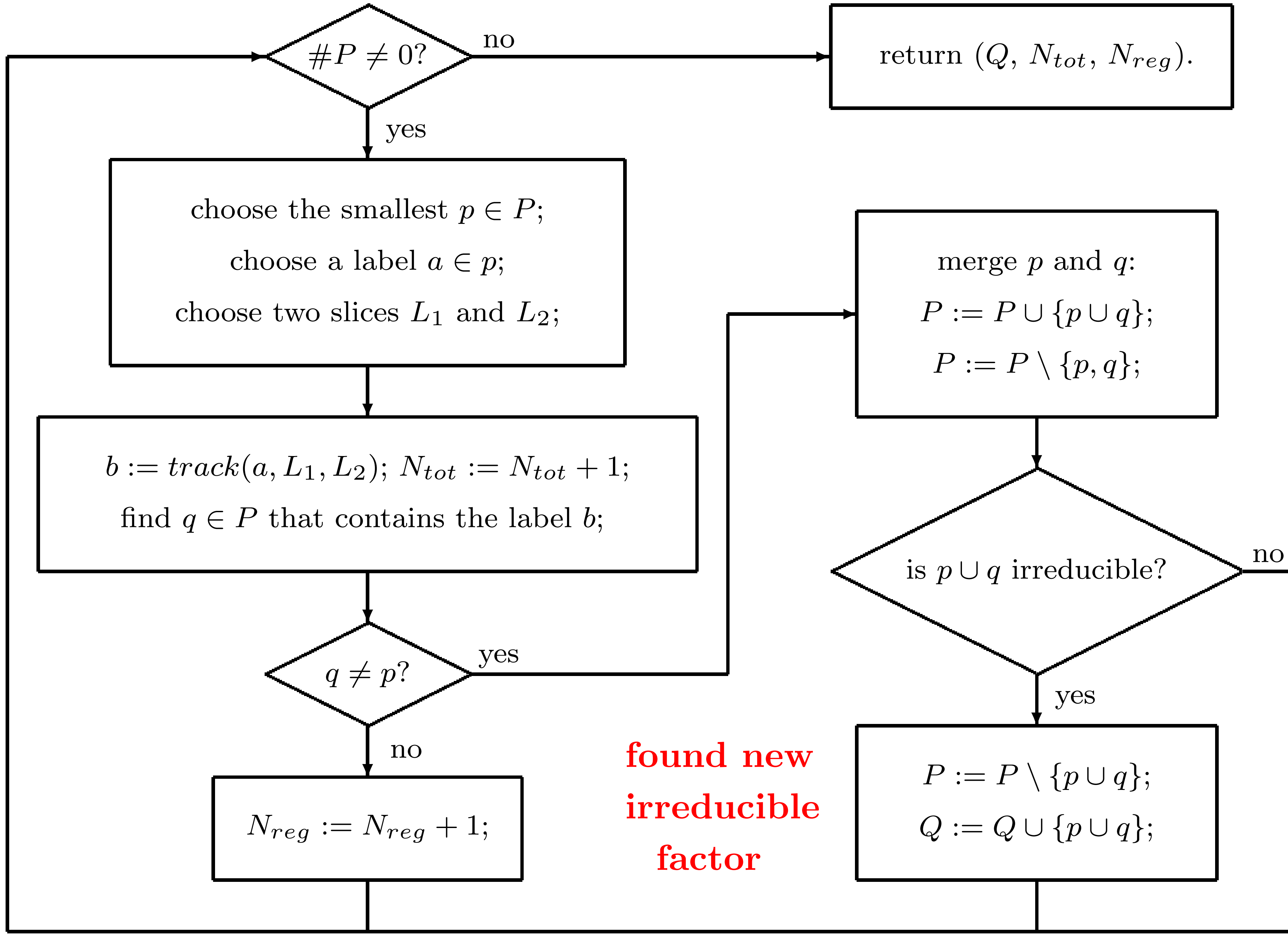

$\#P \neq 0?$
no
return $(Q,\ N_{tot},\ N_{reg})$.
yes
choose the smallest $p \in P$;
choose a label $a \in p$;
choose two slices L_1 and L_2;
merge p and q:
$P := P \cup \{p \cup q\}$;
$P := P \setminus \{p, q\}$;
$b := track(a, L_1, L_2)$; $N_{tot} := N_{tot} + 1$;
find $q \in P$ that contains the label b;
$q \neq p?$
yes
no
$N_{reg} := N_{reg} + 1$;

If a permutation occurred, then p and q are joined.



$\#P \neq 0$?
no
return $(Q,\ N_{tot},\ N_{reg})$.
yes
choose the smallest $p \in P$;
choose a label $a \in p$;
choose two slices L_1 and L_2;
merge p and q:
$P := P \cup \{p \cup q\}$;
$P := P \setminus \{p, q\}$;
$b := track(a, L_1, L_2)$; $N_{tot} := N_{tot} + 1$;
find $q \in P$ that contains the label b;
is $p \cup q$ irreducible?
no
$q \neq p$?
yes
no
$N_{reg} := N_{reg} + 1$;

If the linear trace test fails to certify $p \cup q$, then more loops are needed.


$\#P \neq 0?$
no
return $(Q,\ N_{tot},\ N_{reg})$.
yes
choose the smallest $p \in P$;
choose a label $a \in p$;
choose two slices L_1 and L_2;
$b := track(a, L_1, L_2)$; $N_{tot} := N_{tot} + 1$;
find $q \in P$ that contains the label b;
$q \neq p?$
yes
no
$N_{reg} := N_{reg} + 1$;
merge p and q:
$P := P \cup \{p \cup q\}$;
$P := P \setminus \{p, q\}$;
is $p \cup q$ irreducible?
no
yes
$P := P \setminus \{p \cup q\}$;
$Q := Q \cup \{p \cup q\}$;
found new irreducible factor

The New Algorithm – Parallel Version

Using a manager/worker model:

- Initialization phase involves the distribution of d paths, for the s new witness sets and trace grid.

  Manager distributes jobs to path tracking nodes i, $i > 0$.

- After initialization:

  Manager keeps looking for available nodes to assign paths.

  Other nodes are either busy or ready to start new jobs.

Compared to the first parallel implementation, this algorithm interleaves the computation of the linear trace by the manager with the distribution of path tracking jobs.

Performance on cyclic 8-roots

Five runs using 14 processors (recall 19.7 seconds):

	3 new slices			2 new slices	
#runs	1	2	3	4	5
initial	8.73	9.01	8.89	6.54	6.98
manager	6.06	6.22	6.18	6.67	7.10
min track	5.96	6.16	6.07	6.60	7.02
max track	6.06	6.24	6.23	6.11	7.15
total	14.9	15.4	15.3	13.4	14.2

We report the time used for initialization, the time spent by the manager node, the minimal and maximal time for the nodes spent tracking paths, and the total time, all expressed in seconds.

Conclusions (to appear in IJCSE)

- The new parallel monodromy breakup algorithm shows an even distribution of the time spent by the nodes.
- Using fewer slices reduces initialization time at the expense of a higher running time in main loop.
- Compared to the first parallel implementation, the new algorithm shows a more predictable and regular performance.
- On larger examples, e.g. factoring a 10-dimensional surface of degree 256 in $\mathbb{C}^{18}$, the new algorithm still takes only 80% of the very best time of the first parallel implementation.