Parallel Algorithms for a 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).

The Midwest Algebra, Geometry and their Interactions Conference (MAGIC05)

University of Notre Dame, 8-11 October 2005.

Problem Statement

Given a pure dimensional solution set $f^{-1}(\mathbf{0})$ in $\mathbb{C}^n$,

find its decomposition into irredudible factors.

Special cases:

- When f is a single equation: "Absolute Factorization".
- For ***approximate*** coefficients, almost always irreducible
  $\rightarrow$ find factorization of a polynomial ***close*** to f.

Related Work

Computer Algebra: polynomial factorization is fundamental.

Symbolic-Numeric Computing: **challenge** of Erich Kaltofen (JSC 29, 2000) asks to find a polynomial time algorithm to factor polynomials with **approximate** coefficients.

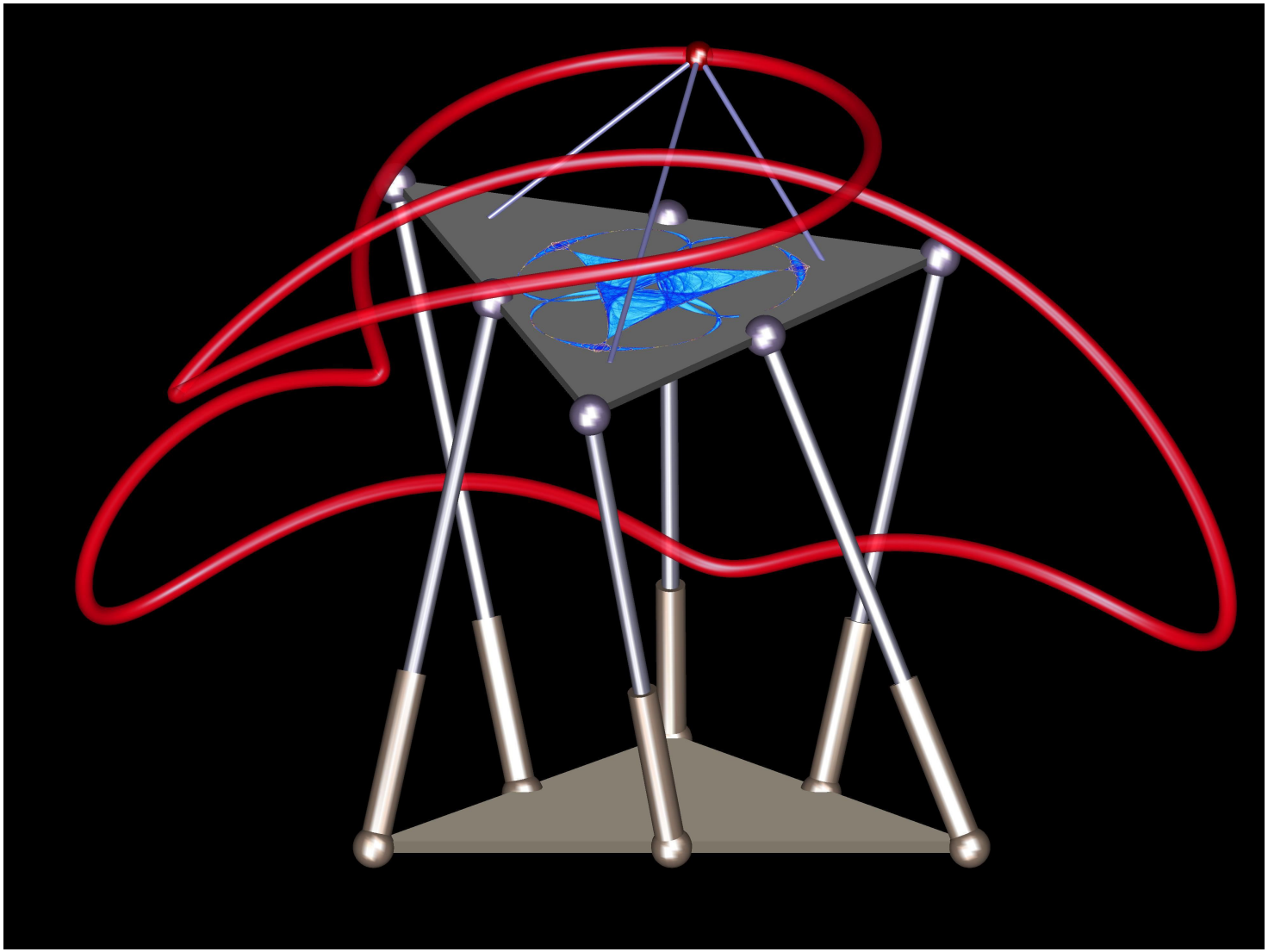
Recent Work: new symbolic-numeric algorithms by

- 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)
- X.-S. Gao, E. Kaltofen, J. May, Z. Yang, L. Zhi (2004)

Numerical Algebraic Geometry:

A.J. Sommese, J. Verschelde, C.W. Wampler (2001,2002).

Irreducible Factor = Assembly of Platform



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!**

Outline of this Talk

Homotopy methods generate loops around singularities
→ **scale very well on parallel computers.**

A first parallel implementation has problems...

A new organization of the loops
→ **leads to a new and more efficient algorithm.**

Computational results: larger problems within reach.

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** $\boldsymbol{h}_{L,K,\boldsymbol{\alpha}}(\mathbf{x},t) = 0$, from $t = 0$ to 1:

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

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

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

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

Linear Traces as Stop Criterium

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

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 $\{(x_1, y_{10}), (x_1, y_{11}), (x_1, y_{12})\}$.

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

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.

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 master node***
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 α; — *broadcast K and α*
   - 4.2 **track d paths** defined by $h_{L,K,\alpha}(\mathbf{x}, t) = \mathbf{0}$; — *executed* ***in parallel***
   - 4.3 generate a random β; — *broadcast β 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 master node***
   - 4.6 **exit when** linear trace test certifies $\mathcal{P}$.

A Benchmark Example: cyclic 8-roots

The system

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

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 master node and 13 slave nodes.**

Computational Results

- 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

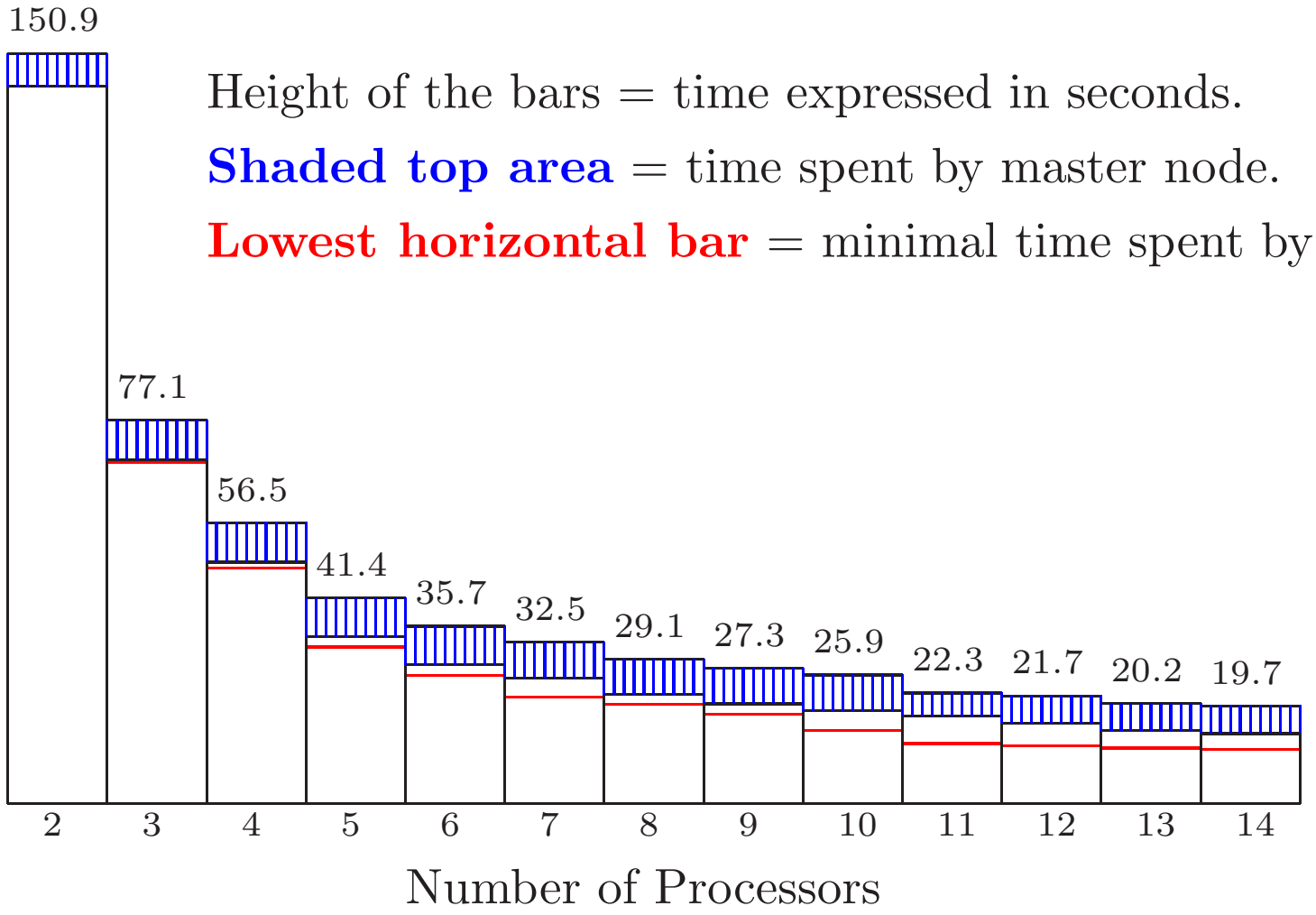
Results of 10 runs on 14 processors. **#L = number of loops**, **min** and **max** are the minimal and maximal time (in seconds) spent by the slave nodes.

- **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 master node.

Lowest horizontal bar = minimal time spent by slave.

The New Algorithm – Serial Version

$P := \{\{a\} \mid 1 \leq a \leq d,\ \{a\} \text{ is not a component}\}$;

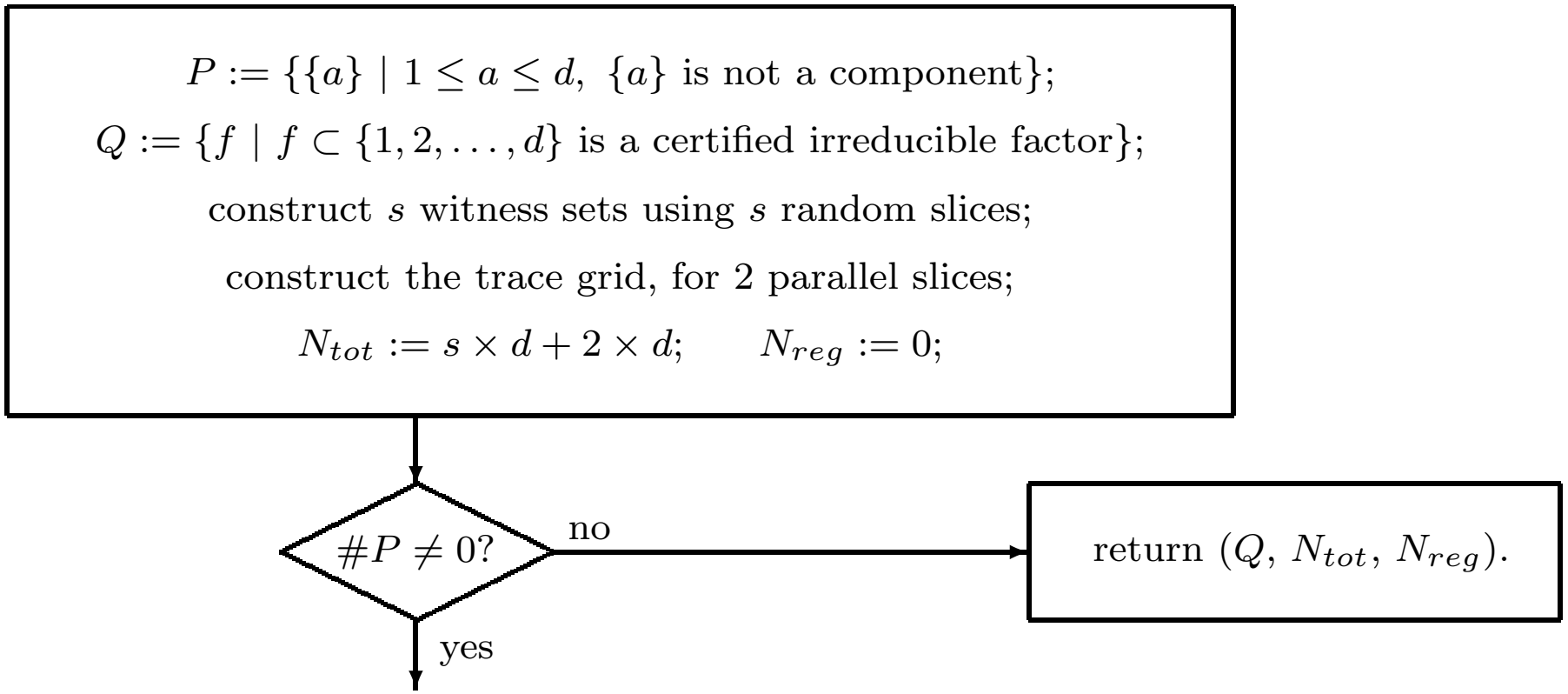
$Q := \{f \mid f \subset \{1, 2, \ldots, d\} \text{ is a certified irreducible factor}\}$;

construct s witness sets using s random slices;

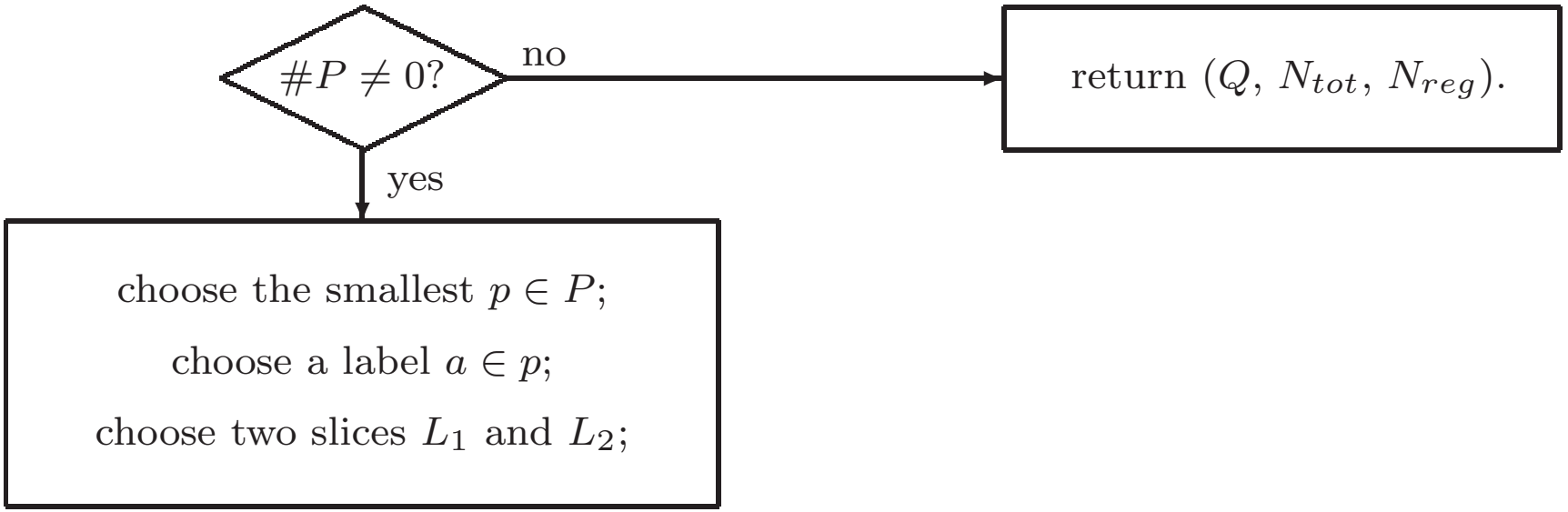
construct the trace grid, for 2 parallel slices;

$N_{tot} := s \times d + 2 \times d$; $\quad N_{reg} := 0$;

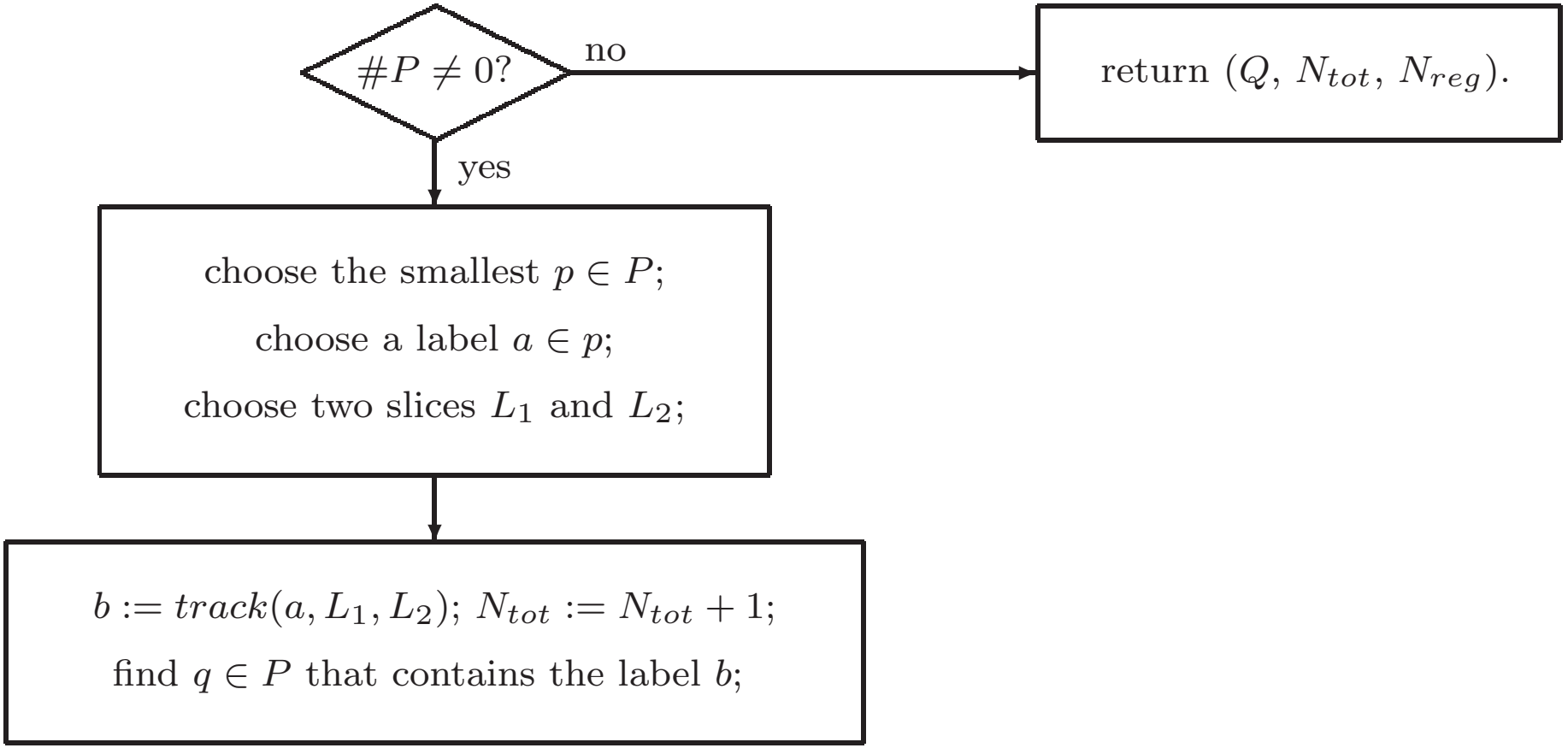
The initialization requires $(s+2)d$ paths



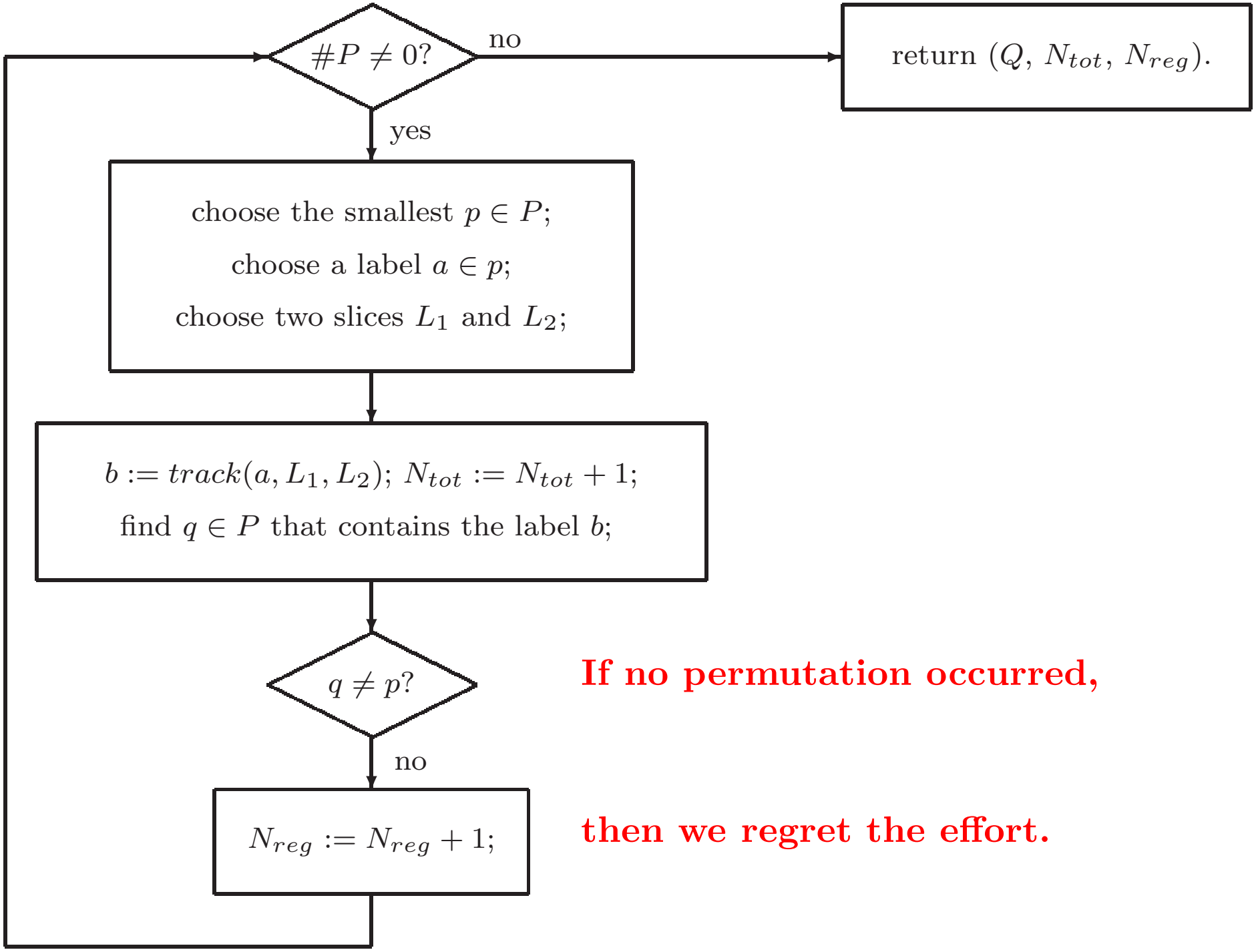
On return is a certified factorization, and path tracking statistics.



Use path tracking statistics to discriminate against nonproductive slices.

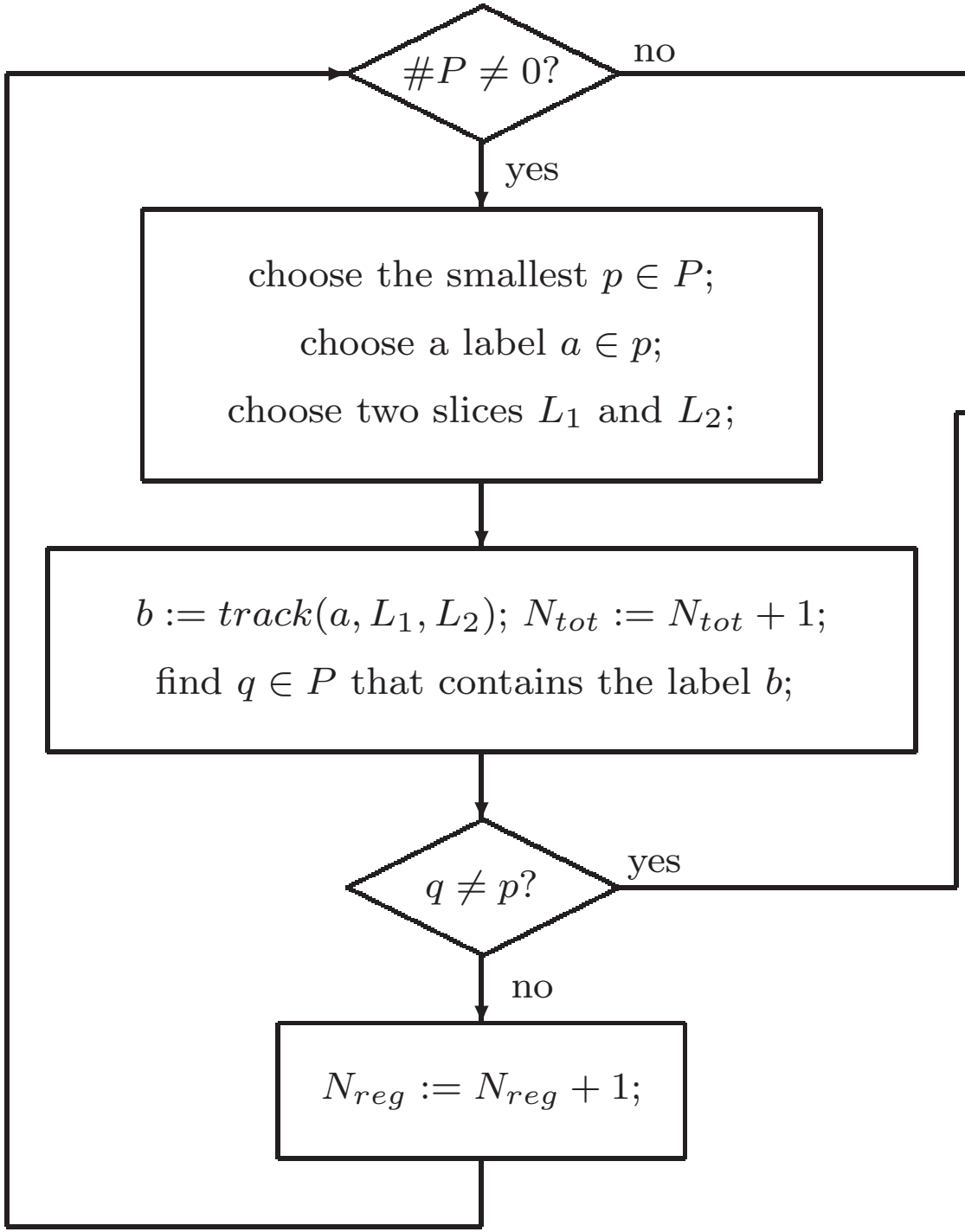


Track a path to close a loop.

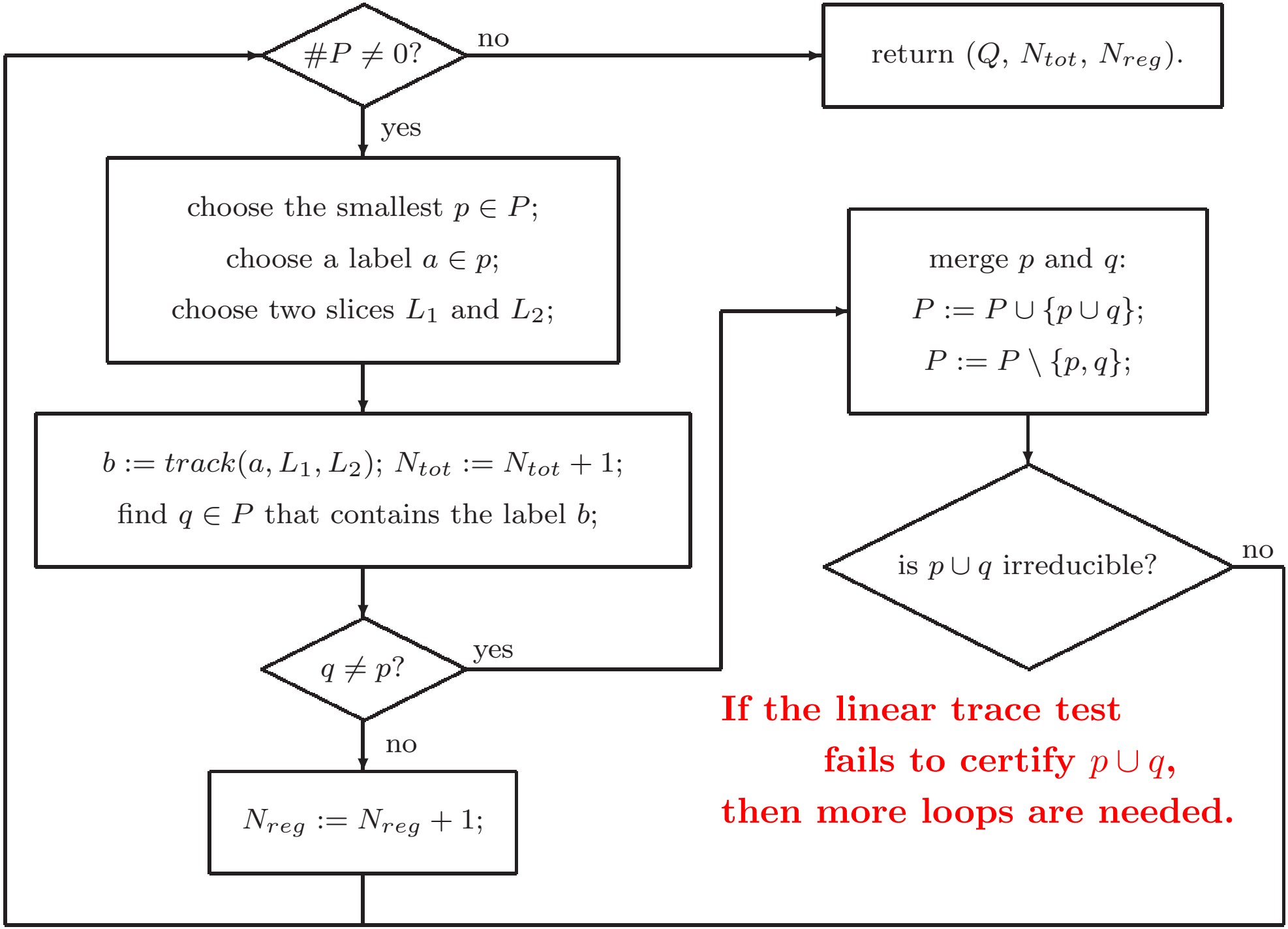


If no permutation occurred,

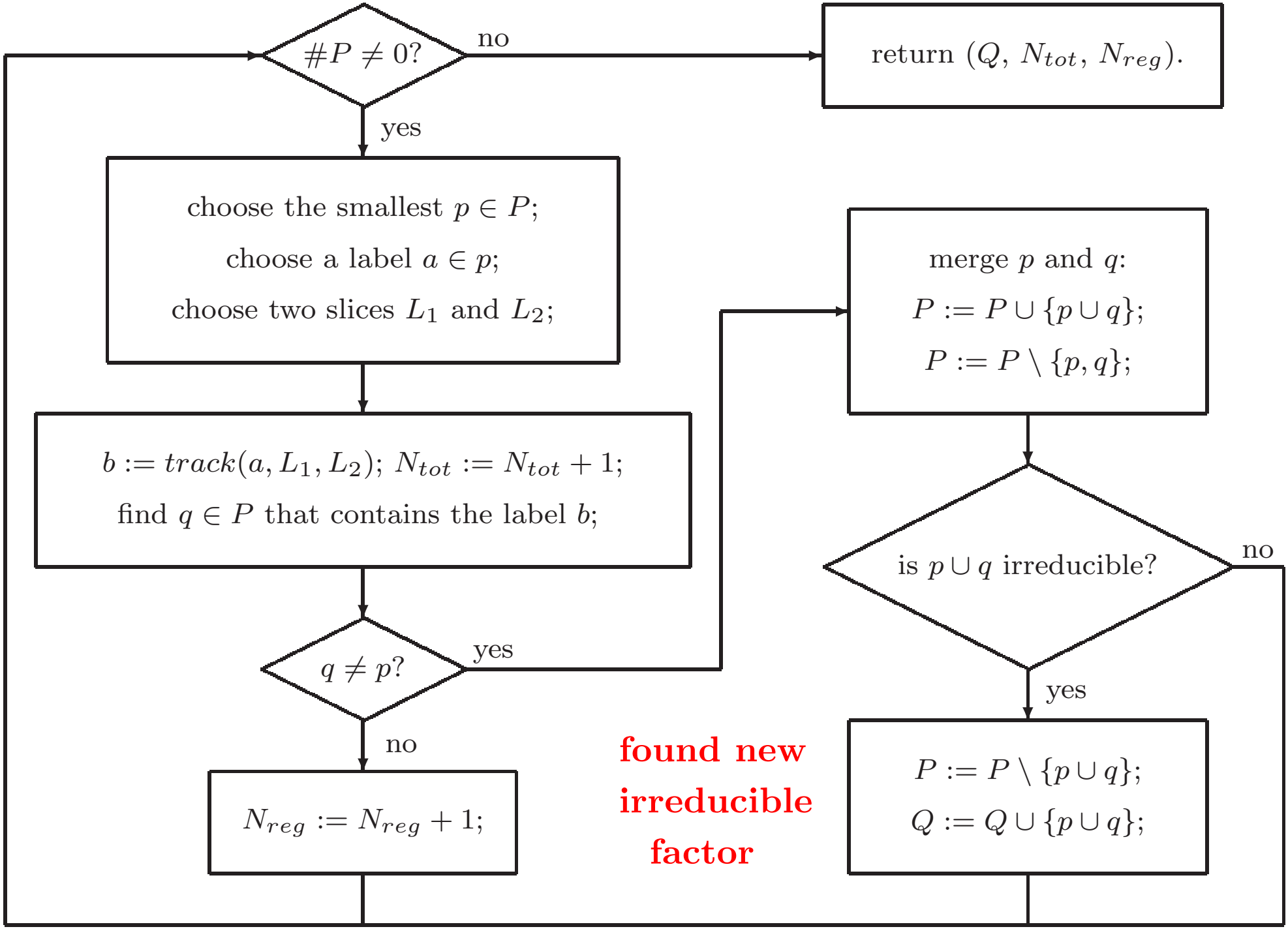
then we regret the effort.



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



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 master/slave model:

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

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

- After initialization:

  Node 0 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 master 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
master	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 master node, the minimal and maximal time for the nodes spent tracking paths, and the total time, all expressed in seconds.

Conclusions

- 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.