Parametric Search \textsuperscript{(Megiddo ‘83)}

A general technique to reduce a problem to a decision problem:

- Compute $t^*$
- Given value $t$, decide whether
  - $t^* < t$
  - $t^* > t$
  - $t^* = t$

Let $T_D =$ time for solving decision problem

If $t^*$ lies in finite universe $U$, can solve orig. problem in

$$O(T_D \log U + U \log U)$$
by binary search.

But what if $U$ is too big?

\textbf{Ex.1} Given $n$ lines in $\mathbb{R}^2$, $l_1, \ldots, l_n$ and integer $k$,

find the $k^{th}$ leftmost intersection point
among $O(n^2)$ intersections.

Naive alg.

$O(n^2)$ time

Better?

Decision Problem: given $t$,

decide whether there are $\geq k$ intersections

to the left of $x = t$

\textit{called slope selection problem (by duality)}

$k = 6$

Reduces to sorting.
reduces to sorting + counting inversions in a permutation.

\[ \implies O(n \log n) \text{ time} \]

But can't apply binary search to solve orig prob.

**Ex2** nearest neighbor search in \( \mathbb{R}^d \)

Decision Problem: given query pt \( q \) & value \( t \), is \( q \) nearest neighbor \( \text{dist} \leq t \)?

i.e. does ball \((q, t)\) contain a pt?

ball range emptiness
red ex to halfspace range emptiness in \( \mathbb{R}^{d+1} \)

\[ O(n^{d/2}) \text{ space, } O((\log n) \text{ query time}) \]

\[ O(n^{1-\frac{d}{2}}) \text{ space, } O(n^{1-\frac{d}{2}}) \text{ query} \]

*Many more exs...*

**The Technique**

let \( d_t(x) \) be the decision algm with \( T_d \) time

idea - simulate \( d_t(x) \) for \( t = t^* \)

but \( t^* \) is not known!

at some pt, \( d_t \) will make a comparison that depends on \( t^* \)

(e.g. is \( b_i \) below \( b_j \) at \( x = t^* \)?)

i.e. \( m_i t^* + b_i \leq m_j t^* + b_j \)

i.e. \( t^* \leq \frac{b_j - b_i}{m_i - m_j} \) (\( m_i > m_j \))

(in CG, comparisons usually reduce to testing signs of const-deg polynomials)
(In CG, comparisons usually reduce to testing signs of const-deg polynomials)

which reduces to comparing $t^*$

with const # of values

How to resolve the comparison?
by calling the decision alg'm

$O(T_D)$ steps to simulate,
each costing $O(T_D)$ time

$\Rightarrow O(T_D^2)$ time

at the end,

get an interval $I$ s.t.

$\forall t \in I$, $\Delta(t)$ has same as $\Delta(t^*)$

$\Rightarrow I$ is $\{t^*\}$.  \qed

Then if decisi. prob can be solved in $T_D$ time,
then orig. problem can be solved in

$O(T_D^2)$ time.

Ex nearest neighbor search,

$O(\sqrt{n/d})$ space, $O(\log n)$ query time.

The Technique Refined:

Suppose there is a parallel decision alg'm $\alpha_{par}$

that requires $T_D$ processors and $T_D$ time

$\Rightarrow O(n)$ processors, $O(\log n)$ time.  in PRAM

(or AKS network)
Idea: Simulate $A_{par}(t)$ for $t = t^*$ at each time step, $A_{par}$ will make $O(T_D)$ comparisons that depend on $t^*$

\[ (e.g. \quad t^* \leq t_1, \quad t^* \leq t_2, \ldots, \quad t^* \leq t_k) \quad \varepsilon = O(T_D) \]

Resolving all $O(T_D)$ terms by $O(\log T_D)$ calls to the decision alg $\hat{A}$ (sequential)

Thus if decision prob. has sequential alg with $T_D$ time & a parallel alg with $T_D$ processes & $T_D$ time

then orig prob. can be solved in

\[ O \left( \left( T_D + T_D \log T_D \right) \cdot T_D \right) \quad \text{time} \]

Ex. Slope selection

\[
\begin{align*}
T_D &= O(n \log n) \\
T_{PD} &= O(n) \\
T_D &= O(\log n)
\end{align*}
\]

\[
\Rightarrow \quad O \left( (n + n \log n \cdot \log n) \cdot \log n \right) = O \left( n \log^3 n \right) \quad \text{time}
\]

Ex. Nearest neighbor \( \tilde{O} (n^{1 - \frac{d}{4d + 2}} \log n) \) query time
Ex
0(n) space, \(\tilde{O}\left(n - \frac{n}{d^{2/3}} \log n\right)\) query time

Rank
Cole '87 improves to
\(O\left((T_D + T_D) \cdot (T_D + \log T_D)\right)\) time
in some cases

(idea - at each step, use \(1\) call to resolve half of comps ..)

\(\Rightarrow O(n \log^2 n)\) time for slope selection

Rank's - only need to parallelize steps that depend on \(T_D\).
Ap nor doesn't need to solve decision problem.
It can decide membership in any finite universe containing \(\mathbb{F}\).

Disadvantages:
- need parallelization
- hard to implement
- extra logs

Simpler alternatives?