Finding Constants in Big-O Notation

Ned Nedialkov

McMaster University
Canada

SE 3F03
March 2013

Outline

Finding constants in $O(n^\alpha)$

- Assume that an algorithm runs in $O(n^\alpha)$, where we do not know α
- We want to find the constants c and α in cn^α
- How to do this?

- Run the algorithm with problem sizes $n_1, n_2, \ldots, n_m$, for some m, for example say $m = 10$
- Time the execution and find $t_1, \ldots, t_m$, where t_i is the CPU time for size n_i
  Note: measure user time, not real time
- We can apply a least-squares fit

- Write $cn_i^\alpha = t_i$, $i = 1, \ldots, m$
- Taking logarithm on both sides, we obtain

$$\log c + \alpha \log n_i = \log t_i, \quad i = 1, \ldots, m$$

- Let $x = \log c$. The above can be written as

$$Ay = \begin{bmatrix} 1 & \log n_1 \\ 1 & \log n_2 \\ \vdots & \\ 1 & \log n_m \end{bmatrix} \begin{bmatrix} x \\ \alpha \end{bmatrix} = \begin{bmatrix} \log t_1 \\ \log t_2 \\ \vdots \\ \log t_m \end{bmatrix} = b$$

- This is an overdetermined system
- It can be solved in Matlab as `y = A\b`
- Then $[x, \alpha] = [y_1, y_2]$ and

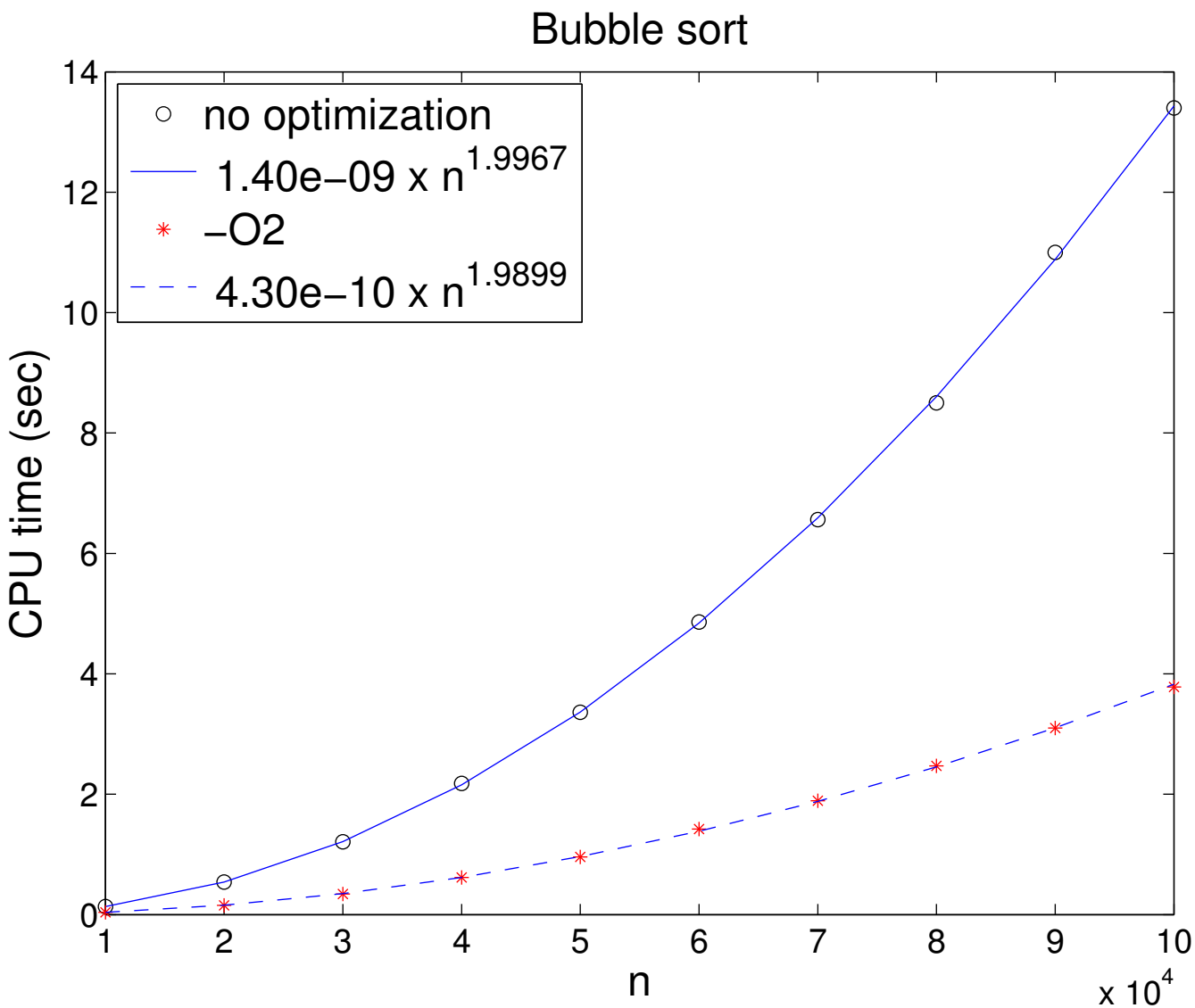
$$c = e^{y_1} \quad \text{and} \quad \alpha = y_2$$

Example

We find such constants for the bubble sort algorithm implemented in

```
void swap( int *a, int *b ) {
  int temp;
  temp = *a; *a = *b; *b = temp;
}
void buble_sort( int *a, int n ) {
  int i, j, max, max_i;
  for ( i = 0; i < n-1; i++ ){
    max = a[i]; max_i = i;
    for ( j = i+1; j < n; j++ )
      if ( a[j] < max ) { max = a[j]; max_i = j;}
    if ( i != max_i )   swap( a+i, a+max_i );
  }
```

- Intel Core i5 1.7 GHz
- 2 cores
- L2 Cache (per Core) 256 KB
- L3 Cache 3 MB
- Memory 4 GB
- gcc compiler


Bubble sort
no optimization
1.40e−09 x n^1.9967
−O2
4.30e−10 x n^1.9899
CPU time (sec)
n
x 10^4
0
2
4
6
8
10
12
14
1
2
3
4
5
6
7
8
9
10