On the Complexity of Register Coalescing

F. Bouchez, A. Darte, and F. Rastello

LIP UMR CNRS — Inria — ENS Lyon — UCBL
France

CGO'07, San Jose

Outline

1. What, Why, and How to Coalesce
   - Basic Formulation
   - The Different Approaches

2. A Hard Optimization Problem

3. Conclusion: What should we Implement Now?

Outline

1 What, Why, and How to Coalesce
- Basic Formulation
- The Different Approaches

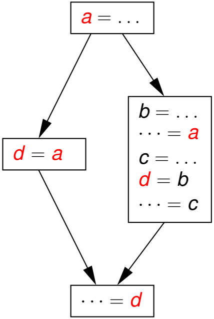
2 A Hard Optimization Problem

3 Conclusion: What should we Implement Now?

Coalescing: Coloring the Interference/Affinity Graph

Coalescing is

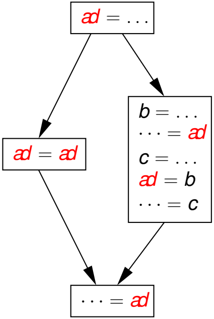
- rename 2 variables into a unique representant
- MOVE A, B: an *affinity* between A and B
- A and B cannot share the same ressource: an *interference* between A and B



Coalescing: Coloring the Interference/Affinity Graph

Coalescing is

- rename 2 variables into a unique representant
- MOVE A, B: an *affinity* between A and B
- A and B cannot share the same ressource: an *interference* between A and B



$ad = \dots$

$ad = ad$

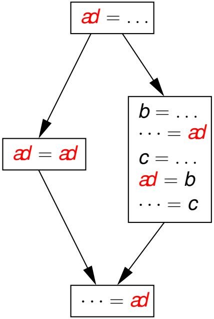
$b = \dots$
$\dots = ad$
$c = \dots$
$ad = b$
$\dots = c$

$\dots = ad$

Coalescing: Coloring the Interference/Affinity Graph

Coalescing is

- rename 2 variables into a unique representant
- MOVE *A*, *B*: an *affinity* between *A* and *B*
- *A* and *B* cannot share the same ressource: an *interference* between *A* and *B*

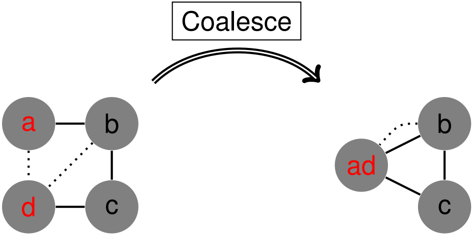


$ad = \dots$

$ad = ad$

$b = \dots$
$\dots = ad$
$c = \dots$
$ad = b$
$\dots = c$

$\dots = ad$

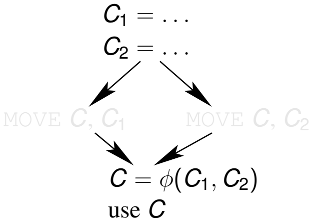


Many MOVE...

Many MOVE instructions due to

- register constraints (function call, 2 address instructions, etc.)
- SSA construction followed by basic SSA destruction

$A = \ldots$
$B = \ldots$
MOVE $R0, A$
MOVE $R1, B$
$D = f(A, B)$
MOVE $D, R0$

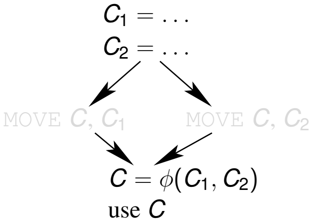


Many MOVE...

Many MOVE instructions due to

- register constraints (function call, 2 address instructions, etc.)
- SSA construction followed by basic SSA destruction

$A = \ldots$
$B = \ldots$
MOVE $R0, A$
MOVE $R1, B$
call f
MOVE $D, R0$



$C_1 = \ldots$
$C_2 = \ldots$

MOVE C, C_1 MOVE C, C_2

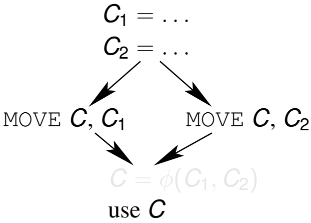
$C = \phi(C_1, C_2)$
use C

Many MOVE...

Many MOVE instructions due to

- register constraints (function call, 2 address instructions, etc.)
- SSA construction followed by basic SSA destruction

```
$A = \dots$
$B = \dots$
MOVE $R0, A$
MOVE $R1, B$
call f
MOVE $D, R0$
```



$C_1 = \dots$
$C_2 = \dots$

MOVE C, C_1

MOVE C, C_2

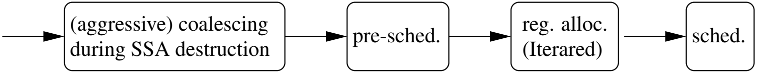
$C = \phi(C_1, C_2)$
use C

Many MOVE... to remove

Many MOVE instructions due to

- register constraints (function call, 2 address instructions, etc.)
- SSA construction followed by basic SSA destruction

Our past experience

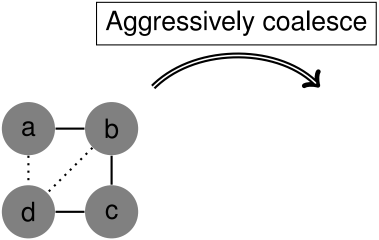


(aggressive) coalescing during SSA destruction → pre-sched. → reg. alloc. (Iterared) → sched.

- on most benchmarks, a good speedup
- on some of them, slow down!

Analysing the slow down...

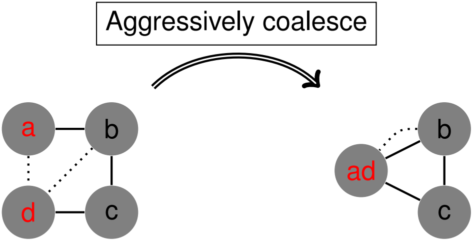
Aggressive coalescing may lead to spilling. Coalescing aware of colorability is *conservative*.



Aggressively coalesce

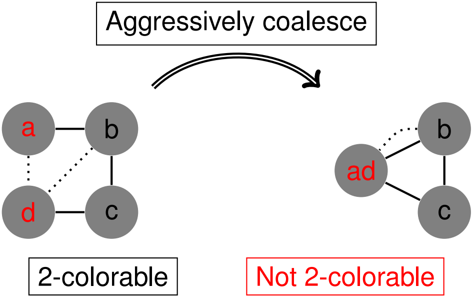
Analysing the slow down...

Aggressive coalescing may lead to spilling. Coalescing aware of colorability is *conservative*.



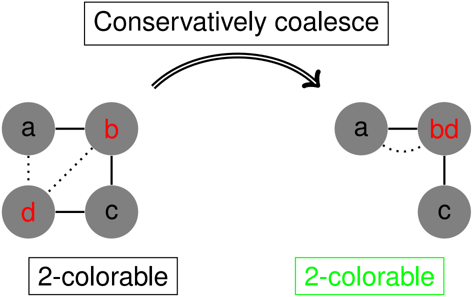
Analysing the slow down...

Aggressive coalescing may lead to spilling. Coalescing aware of colorability is *conservative*.



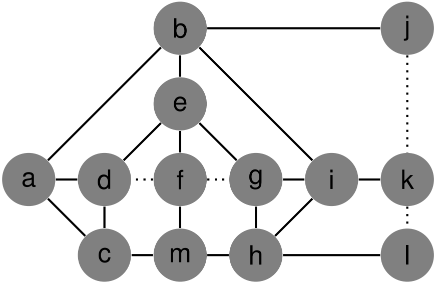
Analysing the slow down...

Aggressive coalescing may lead to spilling. Coalescing aware of colorability is *conservative*.



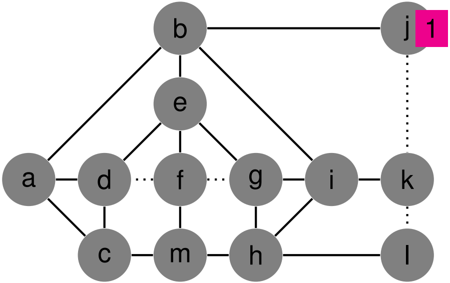
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



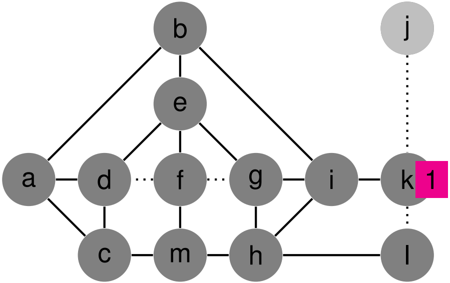
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



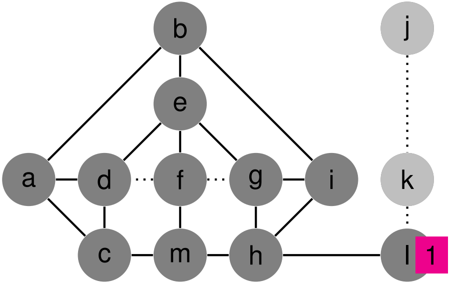
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



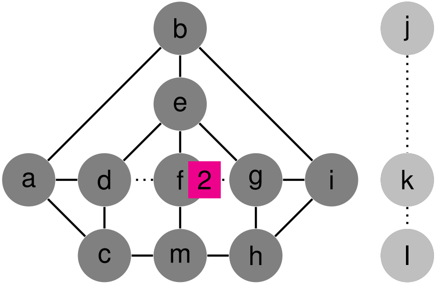
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



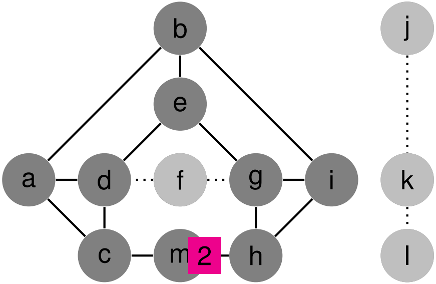
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



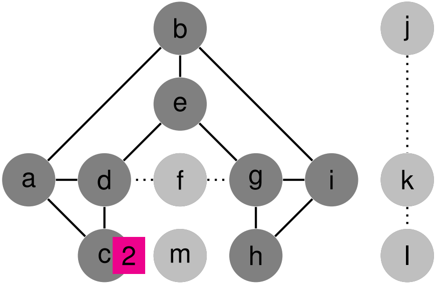
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



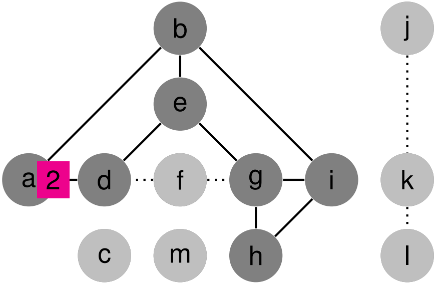
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



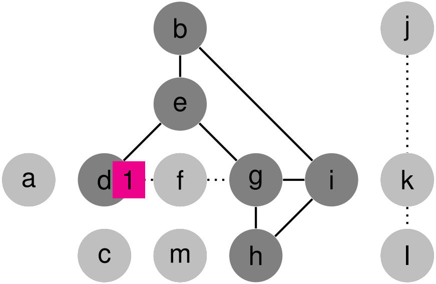
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



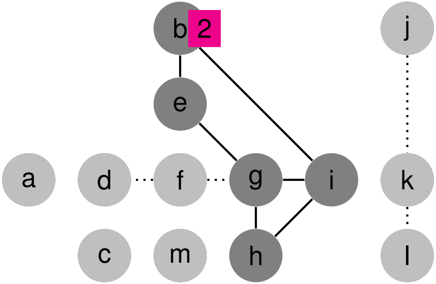
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



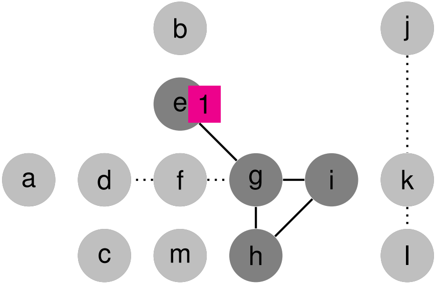
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



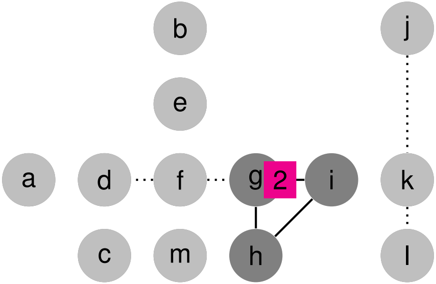
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



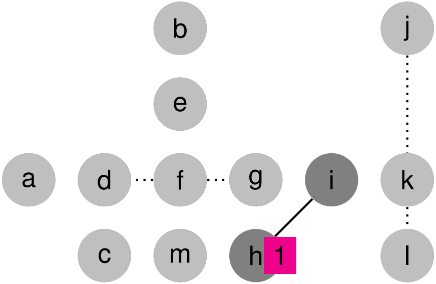
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



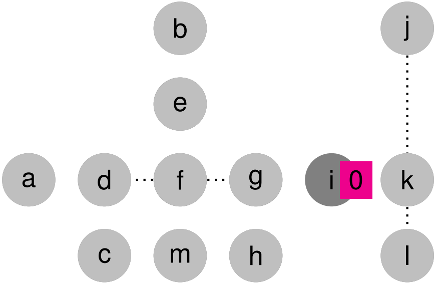
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



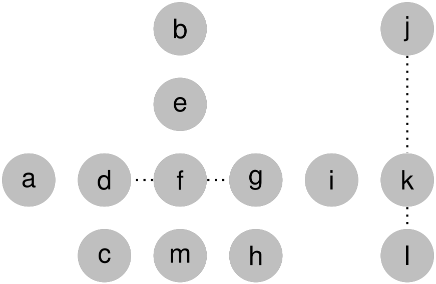
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



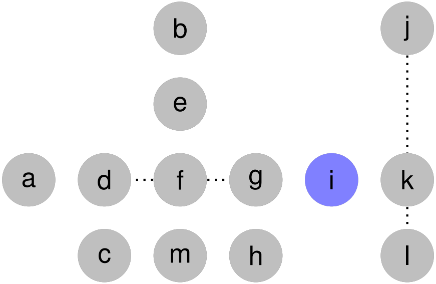
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



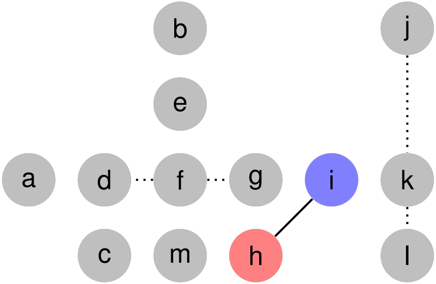
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



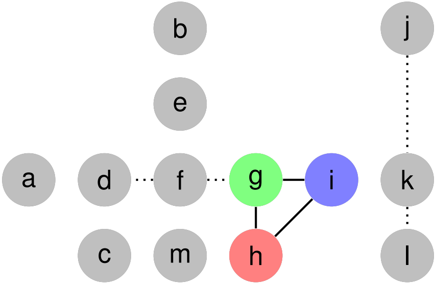
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



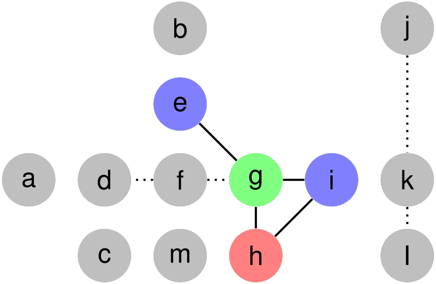
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



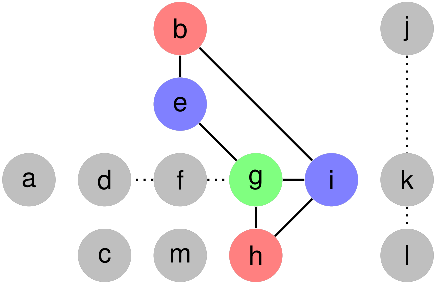
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



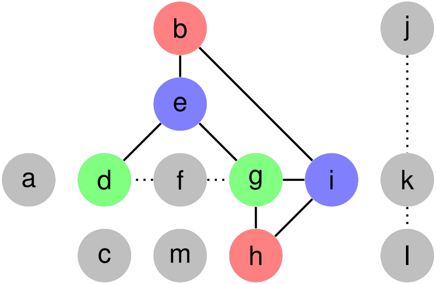
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



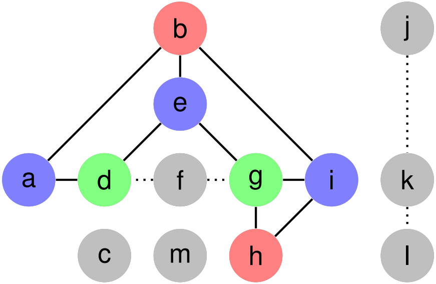
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



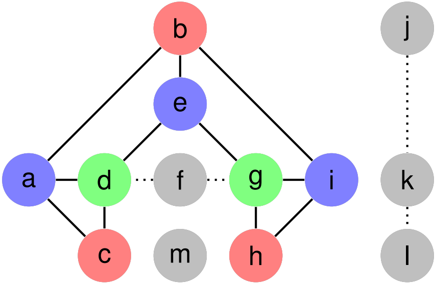
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



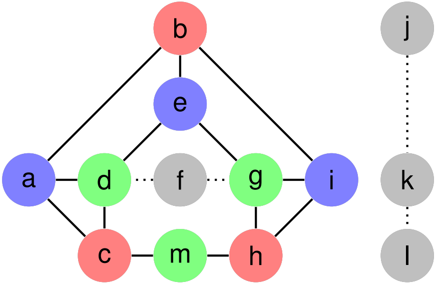
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



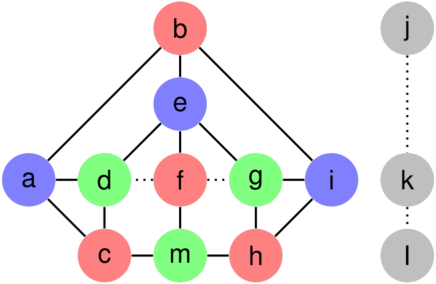
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



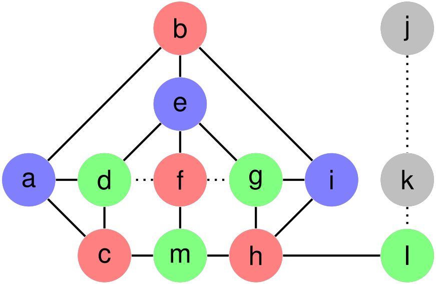
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



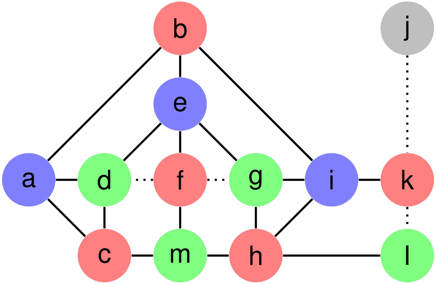
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



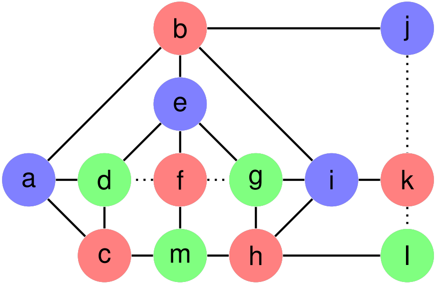
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



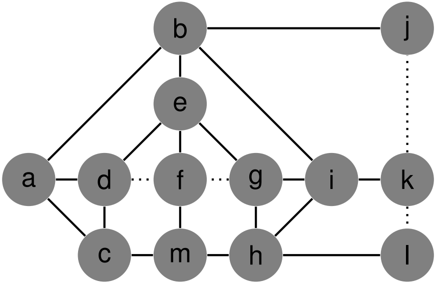
...and the speedup

- *k*-colorability check is hard, but checking the Greedy-k-colorablility is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. *Decoalescing* is better.



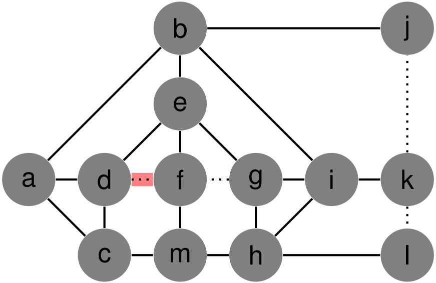
...and the speedup

- *k*-colorability check is hard, but checking the *Greedy-k-colorablility* is easy.
- Still, finding the optimal subset of affinities is hard. We do Incremental coalescing...
- Incremental is not optimal. *Decoalescing* is better.



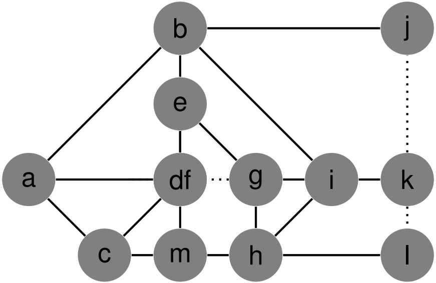
...and the speedup

- *k*-colorability check is hard, but checking the *Greedy-k-colorablility* is easy.
- Still, finding the optimal subset of affinities is hard. We do Incremental coalescing...
- Incremental is not optimal. *Decoalescing* is better.



...and the speedup

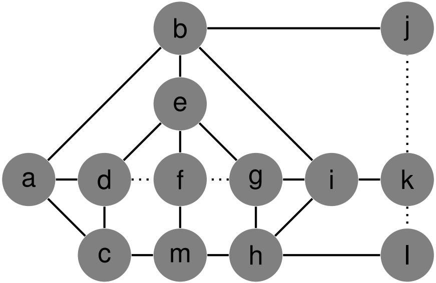
- *k*-colorability check is hard, but checking the *Greedy-k-colorablility* is easy.
- Still, finding the optimal subset of affinities is hard. We do Incremental coalescing...
- Incremental is not optimal. *Decoalescing* is better.



Not greedy-3-colorable

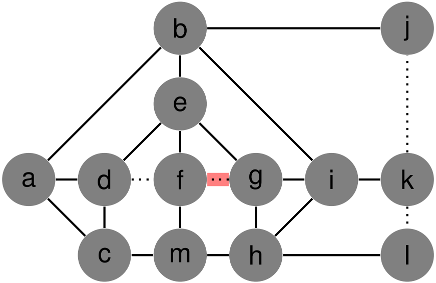
...and the speedup

- *k*-colorability check is hard, but checking the *Greedy-k-colorablility* is easy.
- Still, finding the optimal subset of affinities is hard. We do Incremental coalescing...
- Incremental is not optimal. *Decoalescing* is better.



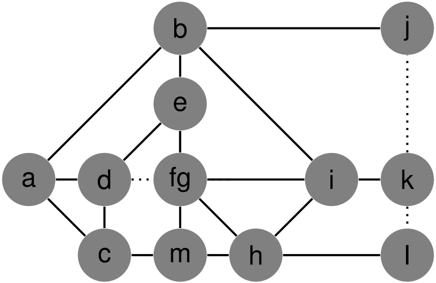
...and the speedup

- *k*-colorability check is hard, but checking the *Greedy-k-colorablility* is easy.
- Still, finding the optimal subset of affinities is hard. We do Incremental coalescing...
- Incremental is not optimal. *Decoalescing* is better.



...and the speedup

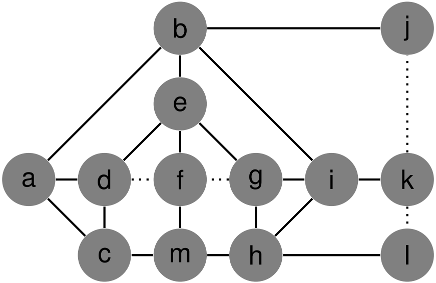
- *k*-colorability check is hard, but checking the *Greedy-k-colorablility* is easy.
- Still, finding the optimal subset of affinities is hard. We do Incremental coalescing...
- Incremental is not optimal. *Decoalescing* is better.



Not greedy-3-colorable

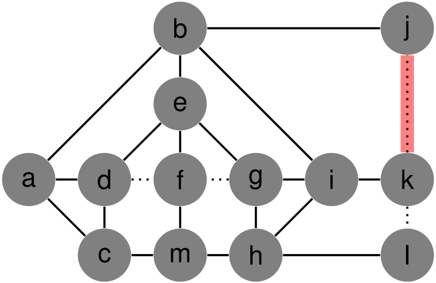
...and the speedup

- *k*-colorability check is hard, but checking the *Greedy-k-colorablility* is easy.
- Still, finding the optimal subset of affinities is hard. We do Incremental coalescing...
- Incremental is not optimal. *Decoalescing* is better.



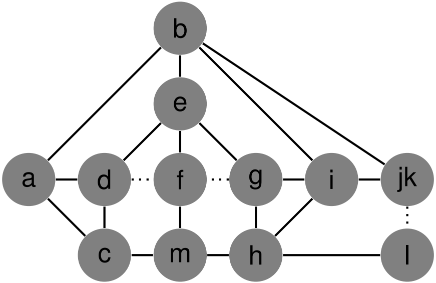
...and the speedup

- *k*-colorability check is hard, but checking the *Greedy-k-colorablility* is easy.
- Still, finding the optimal subset of affinities is hard. We do Incremental coalescing...
- Incremental is not optimal. *Decoalescing* is better.



...and the speedup

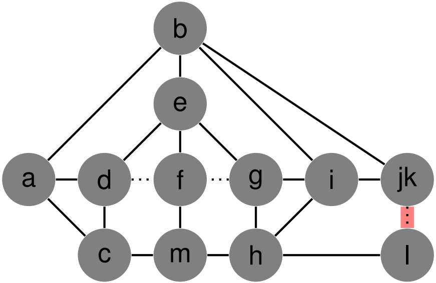
- *k*-colorability check is hard, but checking the *Greedy-k-colorablility* is easy.
- Still, finding the optimal subset of affinities is hard. We do Incremental coalescing...
- Incremental is not optimal. *Decoalescing* is better.



greedy-3-colorable

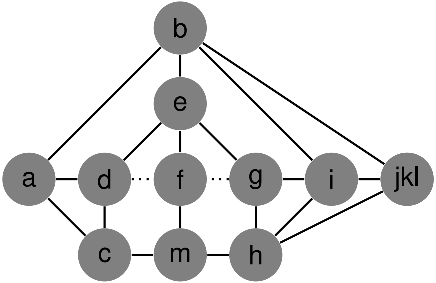
...and the speedup

- *k*-colorability check is hard, but checking the *Greedy-k-colorablility* is easy.
- Still, finding the optimal subset of affinities is hard. We do Incremental coalescing...
- Incremental is not optimal. *Decoalescing* is better.



...and the speedup

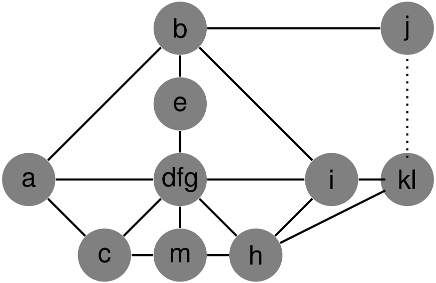
- *k*-colorability check is hard, but checking the *Greedy-k-colorablility* is easy.
- Still, finding the optimal subset of affinities is hard. We do Incremental coalescing...
- Incremental is not optimal. *Decoalescing* is better.



greedy-3-colorable

...and the speedup

- *k*-colorability check is hard, but checking the *Greedy-k-colorablility* is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. Decoalescing is better.

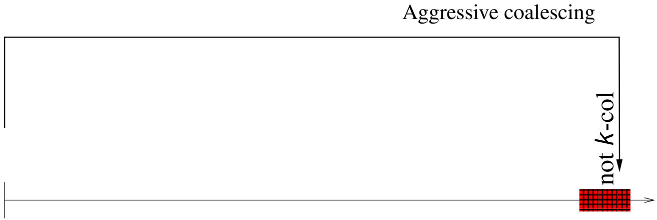


...and the speedup

- *k*-colorability check is hard, but checking the *Greedy-k-colorablility* is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. Decoalescing is better.

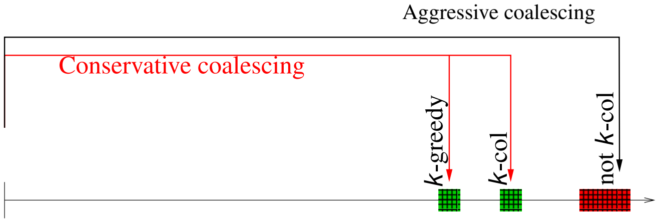
...and the speedup

- *k*-colorability check is hard, but checking the *Greedy-k-colorablility* is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. Decoalescing is better.



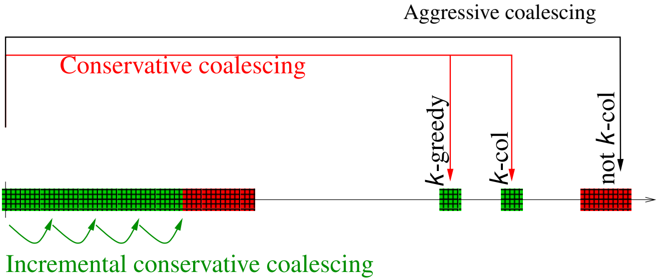
...and the speedup

- *k*-colorability check is hard, but checking the *Greedy-k-colorablility* is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. Decoalescing is better.



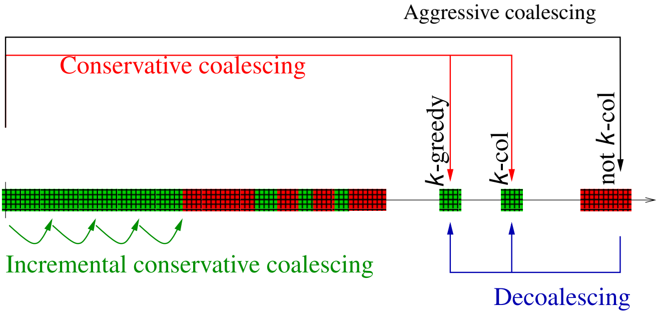
...and the speedup

- *k*-colorability check is hard, but checking the *Greedy-k-colorablility* is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. Decoalescing is better.



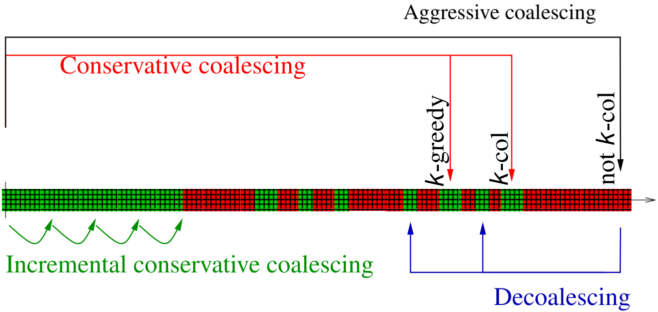
...and the speedup

- *k*-colorability check is hard, but checking the *Greedy-k-colorablility* is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. Decoalescing is better.



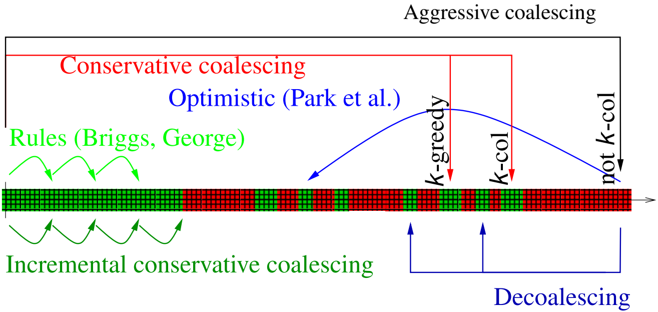
...and the speedup

- *k*-colorability check is hard, but checking the *Greedy-k-colorablility* is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. Decoalescing is better.



...and the speedup

- *k*-colorability check is hard, but checking the *Greedy-k-colorablility* is easy.
- Still, finding the optimal subset of affinities is hard. We do *Incremental* coalescing...
- Incremental is not optimal. Decoalescing is better.



Outline

1 What, Why, and How to Coalesce
- Basic Formulation
- The Different Approaches

2 A Hard Optimization Problem

3 Conclusion: What should we Implement Now?

Coalescing is Hard

G: greedy-*k*-colorable interference graph,

Aggressive coalescing NP-complete, even with $k = 3$.

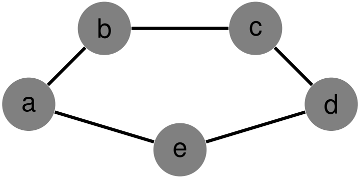
Conservative coalescing NP-complete even if $k = 3$ and only affinities can be merged.

Incremental conservative coalescing (Briggs, George)
NP-complete if *G* is arbitrary.
Open if *G* is greedy-*k*-colorable.

Optimistic coalescing (Park & Moon) = conservative de-coalescing
NP-complete even if $k = 4$.

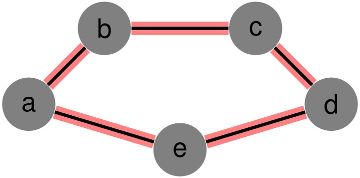
A little hope: Chordal Graphs

- Interference graph of SSA programs
- k-colorability easy on chordal graphs
- $\textsc{Maxlive} = w(G) = \chi(G)$
- k-chordal $\subset$ greedy-k-colorable $\subset$ k-colorable



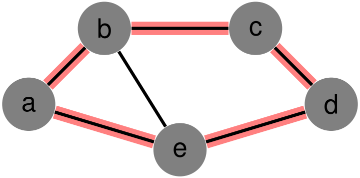
A little hope: Chordal Graphs

- Interference graph of SSA programs
- k-colorability easy on chordal graphs
- MAXLIVE $= w(G) = \chi(G)$
- k-chordal $\subset$ greedy-k-colorable $\subset$ k-colorable



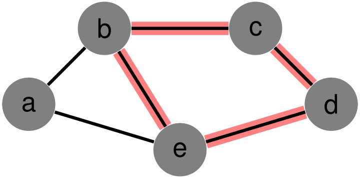
A little hope: Chordal Graphs

- Interference graph of SSA programs
- *k*-colorability easy on chordal graphs
- MAXLIVE $= w(G) = \chi(G)$
- k-chordal $\subset$ greedy-k-colorable $\subset$ k-colorable



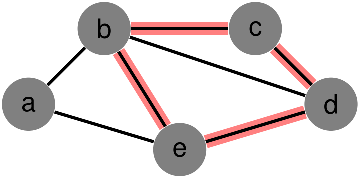
A little hope: Chordal Graphs

- Interference graph of SSA programs
- k-colorability easy on chordal graphs
- $\textsc{Maxlive} = w(G) = \chi(G)$
- k-chordal $\subset$ greedy-k-colorable $\subset$ k-colorable



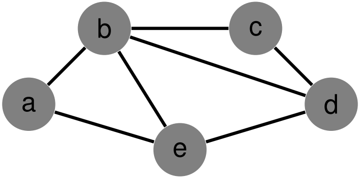
A little hope: Chordal Graphs

- Interference graph of SSA programs
- k-colorability easy on chordal graphs
- $\textsc{MaxLive} = w(G) = \chi(G)$
- k-chordal $\subset$ greedy-k-colorable $\subset$ k-colorable



A little hope: Chordal Graphs

- Interference graph of SSA programs
- k-colorability easy on chordal graphs
- $\textsc{Maxlive} = w(G) = \chi(G)$
- k-chordal $\subset$ greedy-k-colorable $\subset$ k-colorable



. . . but Coalescing is still Hard

G: *k*-chordal interference graph.

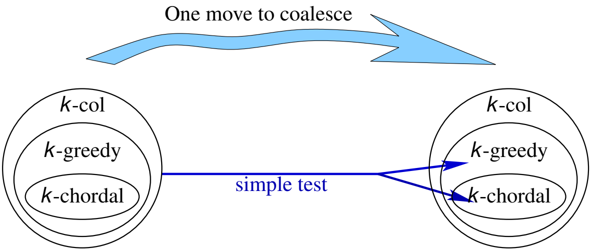
Aggressive coalescing NP-complete.

Conservative coalescing NP-complete.

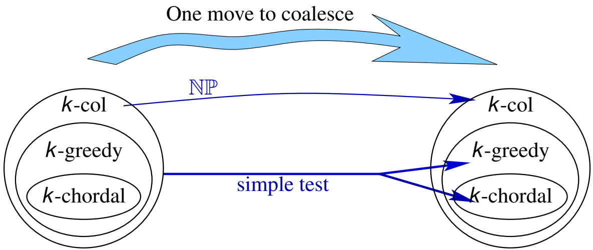
Incremental conservative coalescing (Briggs, George)
Polynomial!

Optimistic coalescing (Park & Moon) = conservative de-coalescing
NP-complete even if $k = 4$.

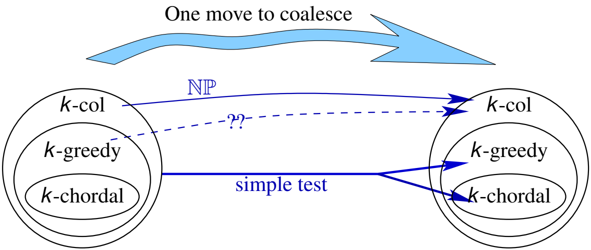
"Multiple-move" incremental



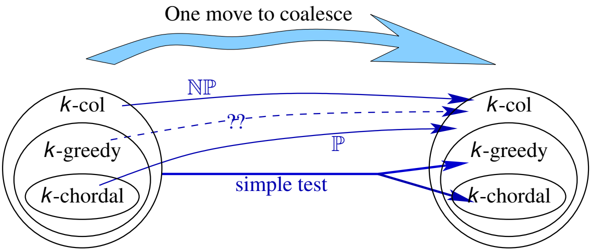
“Multiple-move” incremental



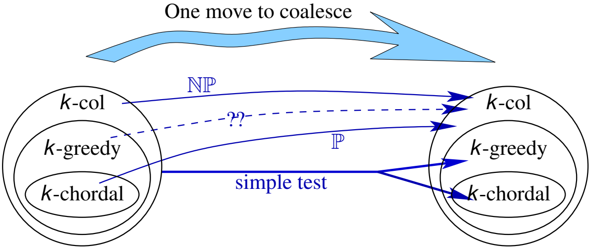
“Multiple-move” incremental

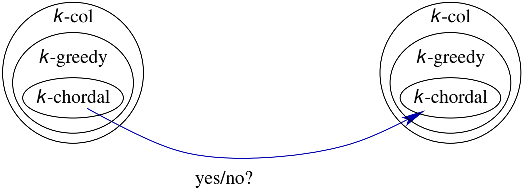


"Multiple-move" incremental

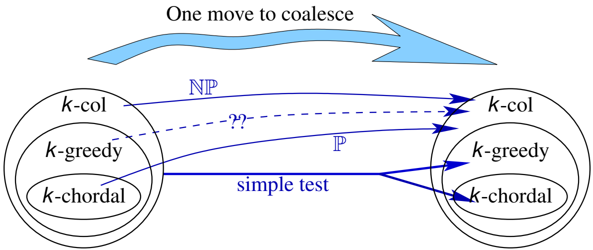


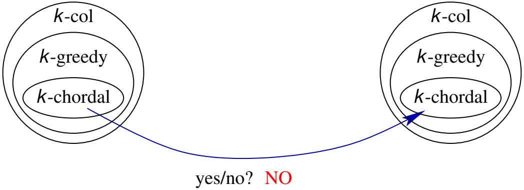
“Multiple-move” incremental



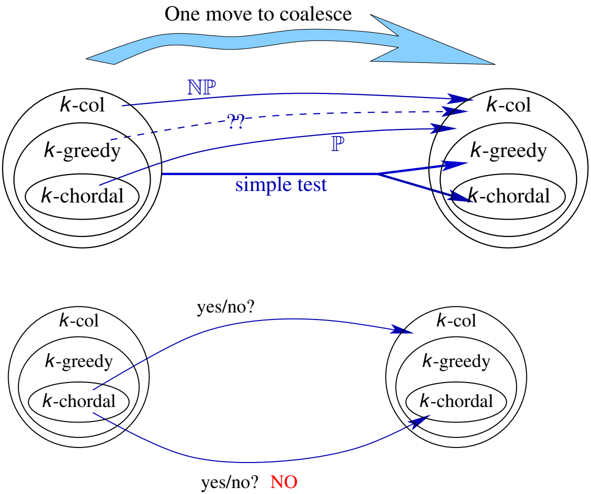


“Multiple-move” incremental

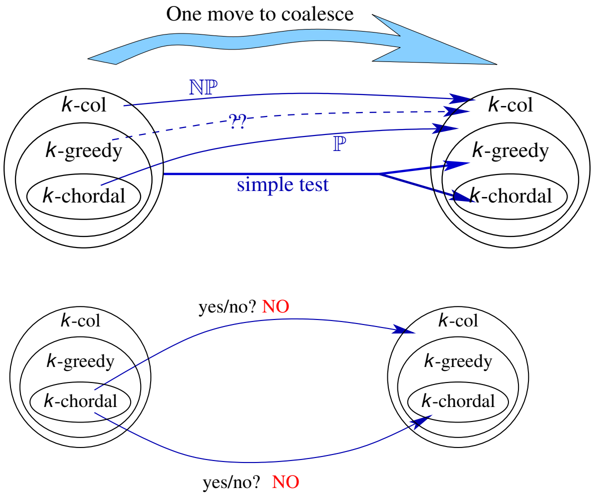




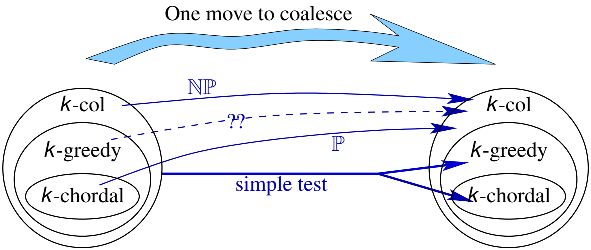
“Multiple-move” incremental

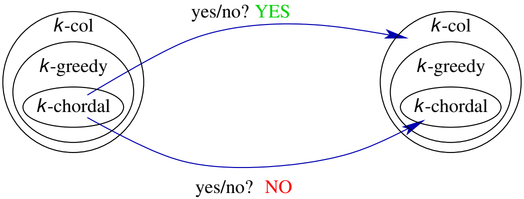


“Multiple-move” incremental

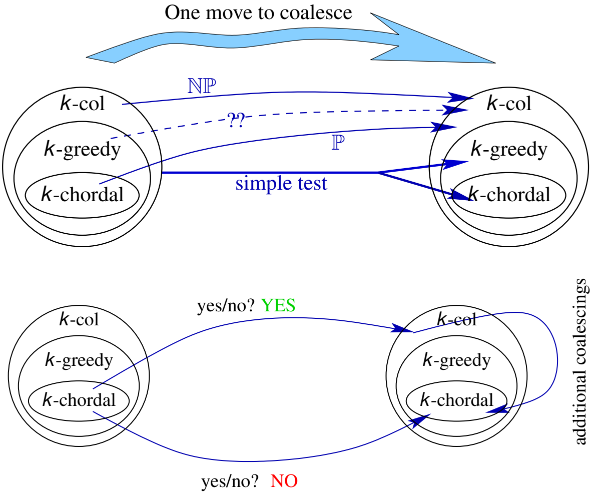


“Multiple-move” incremental





"Multiple-move" incremental



Outline

1 What, Why, and How to Coalesce

- Basic Formulation
- The Different Approaches

2 A Hard Optimization Problem

3 Conclusion: What should we Implement Now?

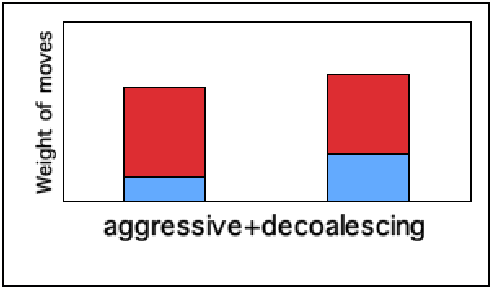
Some measurements

Aggressive+decoalescing scheme:

- optimizing the aggressive part is important
- decoalescing (optimistic) can still be improved

Incremental scheme:

- conservative rules (Briggs, George) are far from the optimal



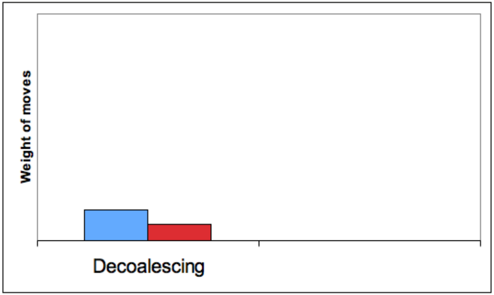
Some measurements

Aggressive+decoalescing scheme:

- optimizing the aggressive part is important
- decoalescing (optimistic) can still be improved

Incremental scheme:

- conservative rules (Briggs, George) are far from the optimal



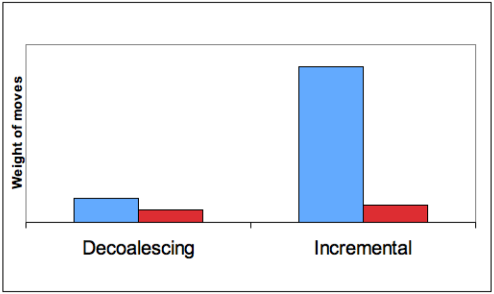
Some measurements

Aggressive+decoalescing scheme:

- optimizing the aggressive part is important
- decoalescing (optimistic) can still be improved

Incremental scheme:

- conservative rules (Briggs, George) are far from the optimal



Conclusion

Most problems are NP-complete $\Rightarrow$ heuristics!

Aggressive+decoalescing scheme:

- Aggressive coalescing is an important issue!
- Still gap for improving decoalescing;

Incremental scheme:

- A large gap for incremental;
- Promizing approach: multiple-move incremental on *k*-greedy.