Implémentation de l'algorithme de Dinic et d'Edmonds-Karp

TP Algorithmique, Complexité & Calculabilité (FMIN105)

William Dyce Thibaut Marmin
Clément Sipieter

Université Montpellier 2

15 Décembre 2011

TP Algorithmique, Complexité & Calculabilité

TP Algorithmique, Complexité & Calculabilité

Algorithmes

Edmonds-Karp

Algorithmes

Dinic

TP Algorithmique, Complexité & Calculabilité

Choix du langage de programmation

C++

- rapidité d'éxecution
- langage à objets
- connaissance du langage
- langage très répandu

Représentation du problème de flot maximum

Réseau de transport

- Graphe orienté et pondéré
- une source
- un puits

Graphe d'écarts

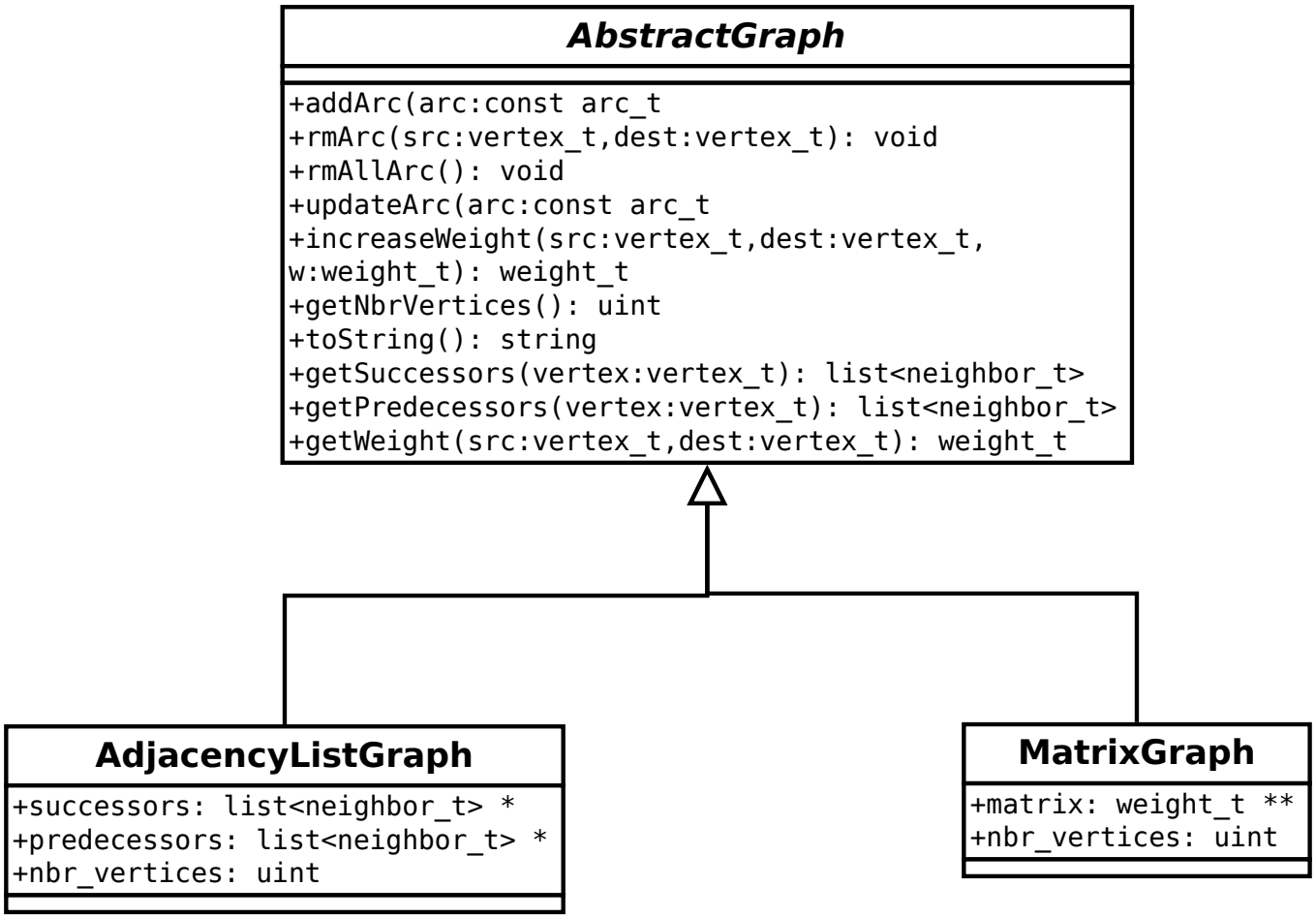
- Graphe orienté et pondéré

Représentation du problème de flot maximum

Graphe de couches

- Graphe orienté et pondéré
- un tableau de listes de sommets (chaque liste représente une couche)

Diagramme de classes



Génération de réseaux de transport aléatoires

Deux stratégies

- Tirage aléatoire de deux sommets
- Génération de tous les arcs possibles et tirage d'un arc

Edmonds-Karp

- Recherche du plus court chemin en nombre d'arcs par un parcours en largeur

Dinic

- Génération du graphe de couche par un parcours en largeur en stockant l'ensemble des parents.
- Complexité de l'accès au prédécesseur dans le graphe de couche pour le calcul du flot bloquant

TP Algorithmique, Complexité & Calculabilité

Méthode de tests

Série de tests

Complexité

- Edmonds-Karp : $O(nm^2)$
- Dinic : $O(n^2m)$

Tests effectués

- Nombre de sommets : 100, 200, 300, ...1000
- Couverture du graphe : 10%, 20%, 30%, ...100%

Méthode de tests

Profiling

GNU gprof

Profiler, analyse du code en fonction du temps passé par chaque fonction à l'exécution.

- Compilation avec l'argument `-pg`
- Exécution du programme, génération du fichier `gmon.out`
- Exportation des statistiques en fichier texte

Tests & résultats

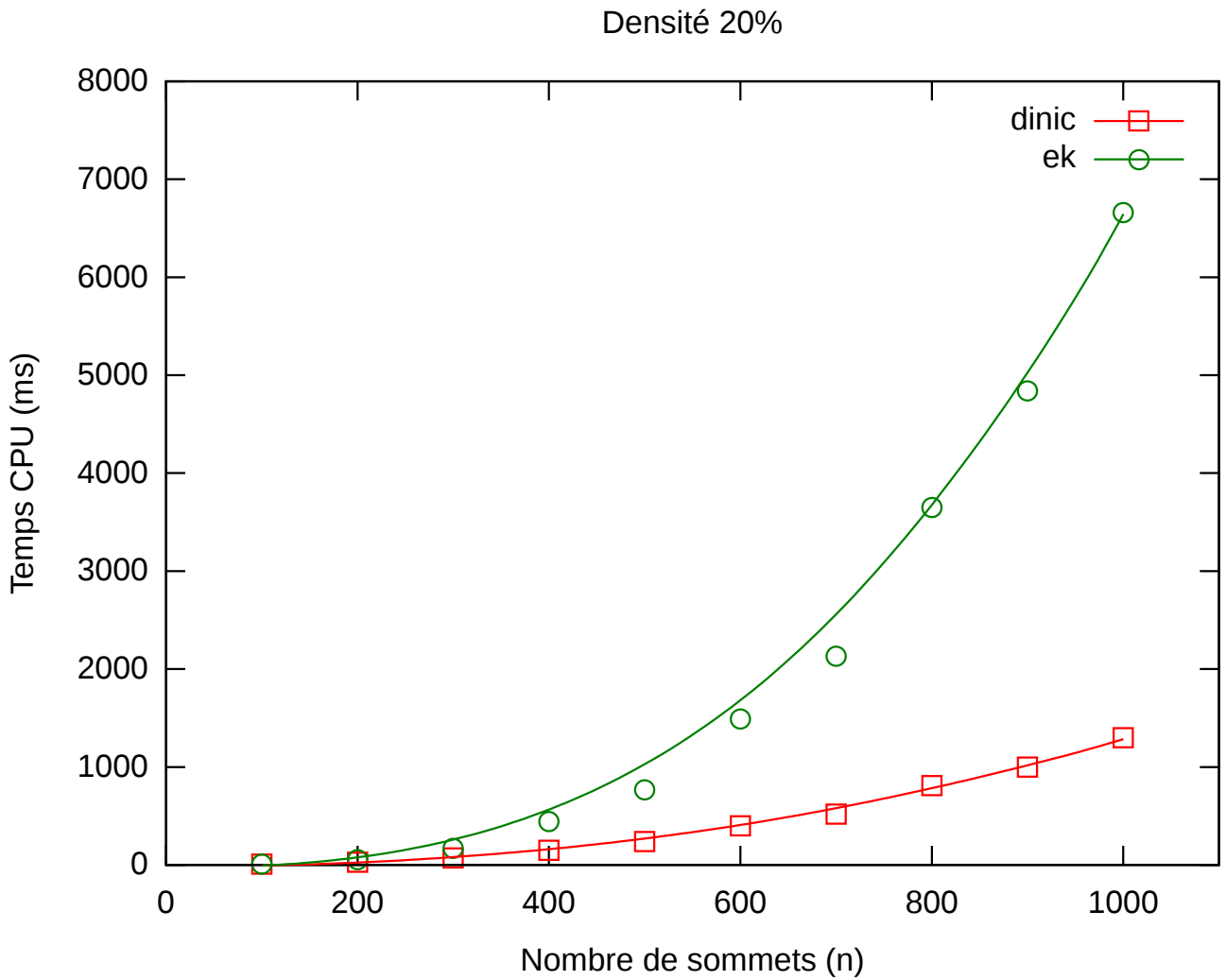
Profiling

GNU gprof

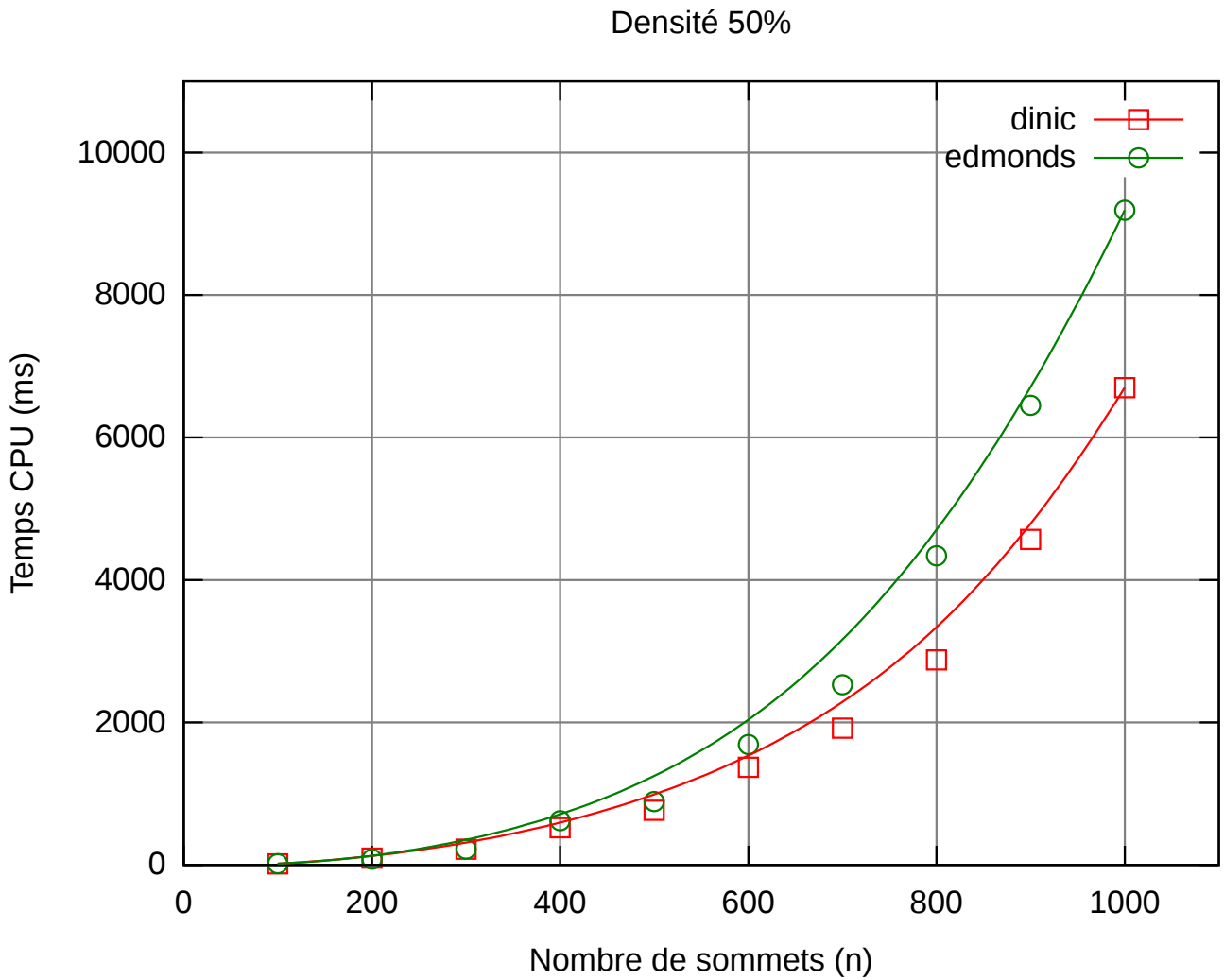
Statistique fournies pour chaque fonction :

- % temps cpu total
- temps cpu
- temps cpu par appel (de manière cumulative ou non)

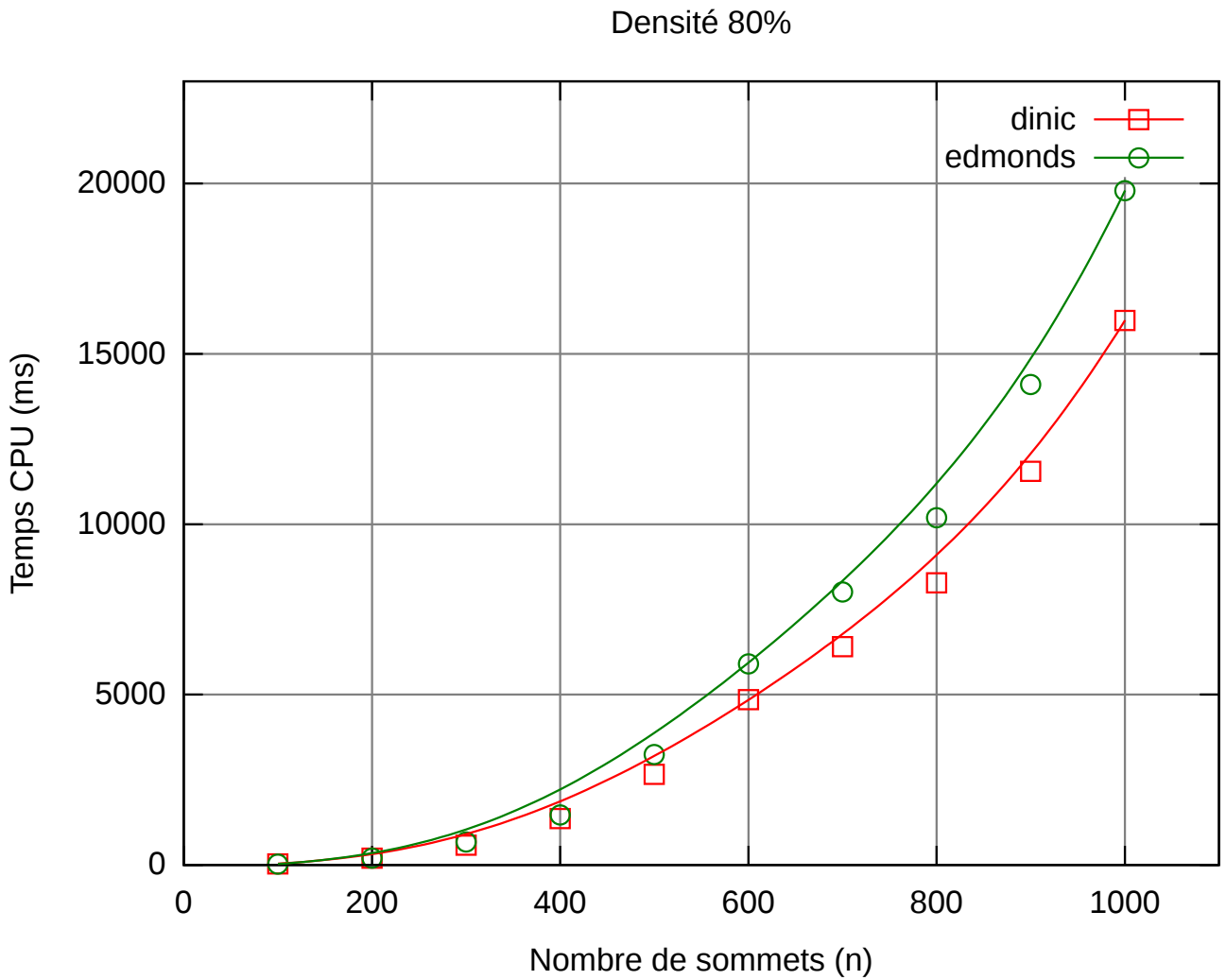
Résultats


Densité 20%
dinic
ek
Temps CPU (ms)
Nombre de sommets (n)
0
1000
2000
3000
4000
5000
6000
7000
8000
0
200
400
600
800
1000

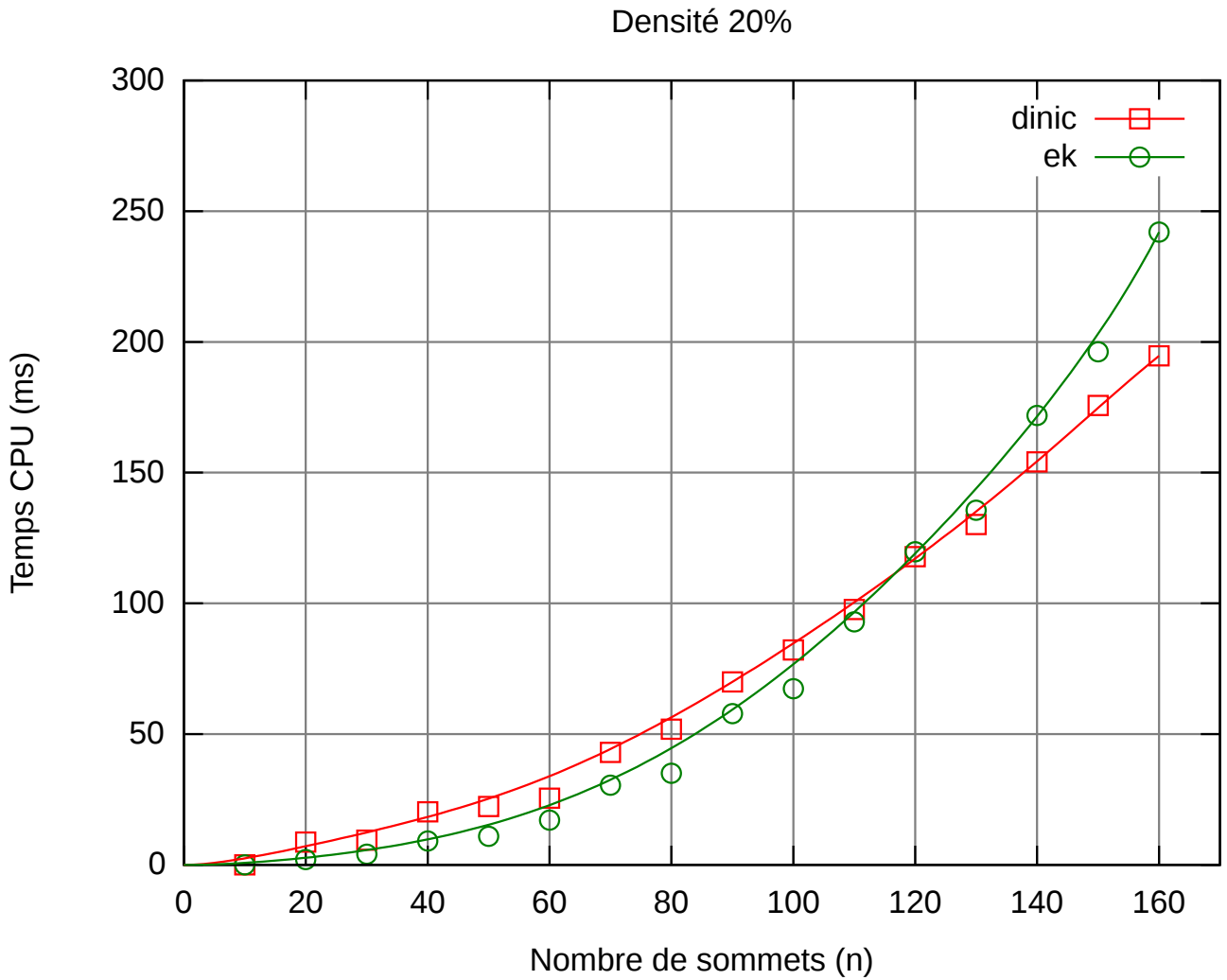
Résultats


Densité 50%
dinic
edmonds
Temps CPU (ms)
Nombre de sommets (n)
0
2000
4000
6000
8000
10000
0
200
400
600
800
1000

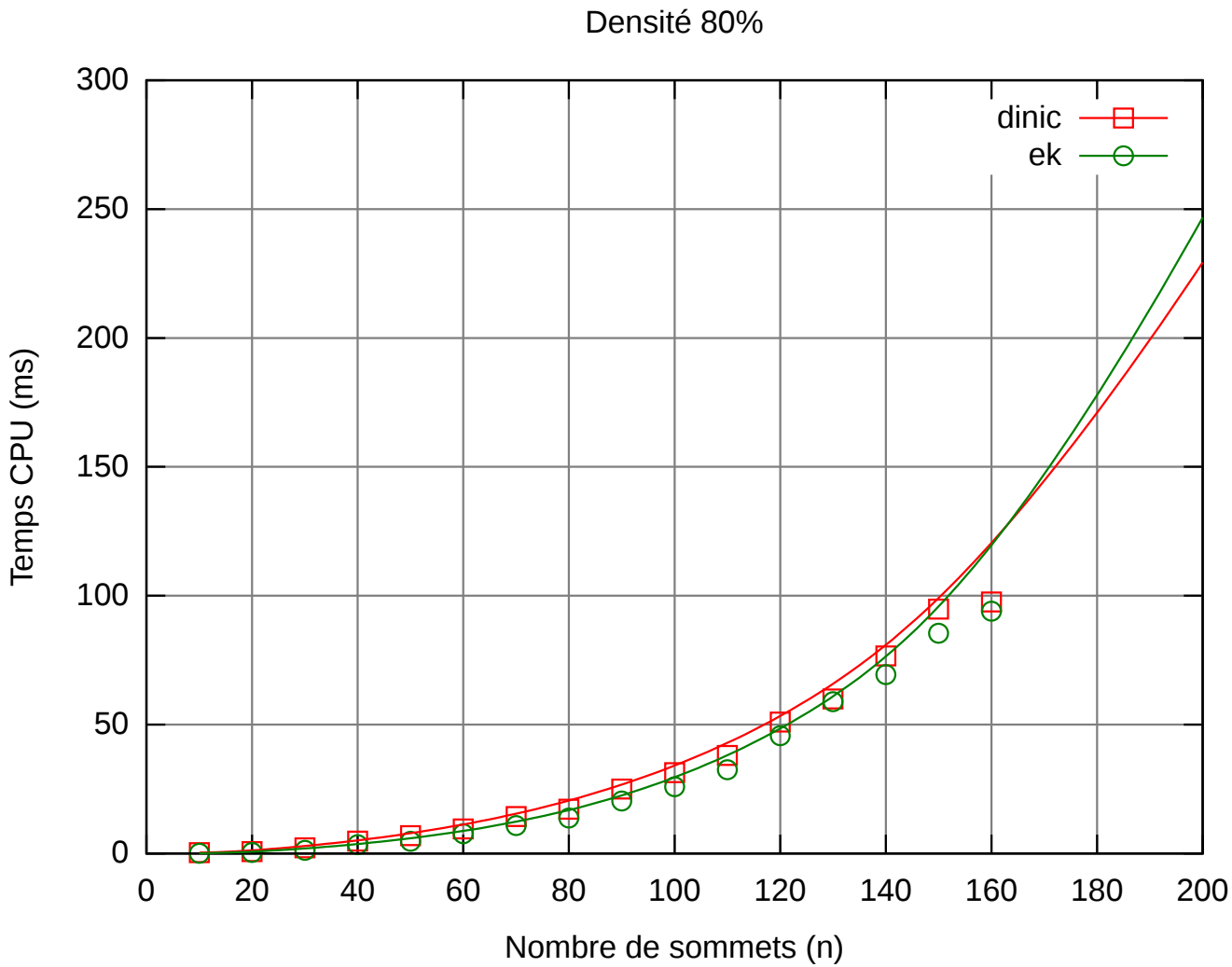
Résultats


Densité 80%
dinic
edmonds
Temps CPU (ms)
Nombre de sommets (n)
0
5000
10000
15000
20000
0
200
400
600
800
1000

Résultats


Densité 20%
dinic
ek
Temps CPU (ms)
Nombre de sommets (n)
0
50
100
150
200
250
300
0
20
40
60
80
100
120
140
160

Résultats


Densité 80%
dinic
ek
Temps CPU (ms)
Nombre de sommets (n)
0
50
100
150
200
250
300
0
20
40
60
80
100
120
140
160
180
200

Résultats

Conclusion

- Dinic globalement plus rapide
- Surtout sur des graphes peu denses
- Edmonds-Karp efficace sur des petits graphes

$\Rightarrow$ Cohérence avec les complexités théoriques

Dinic
$O(nm^2)$

Edmonds-Karp
$O(n^2m)$

TP Algorithmique, Complexité & Calculabilité

Conclusion

TP Algorithmique, Complexité & Calculabilité

Démonstration