

Dans cette thèse, nous nous intéressons essentiellement aux algorithmes de dessin automatique des graphes que nous appellerons **algorithmes d'agencement** des graphes. La performance de ces algorithmes est testée sur la base de la qualité du dessin (ou de l'agencement) qu'ils produisent et du temps d'exécution nécessaire pour le faire.

Trois classes de graphes sont considérés dans cette thèse. Notre étude est organisée suivant le type de graphes que l'algorithme accepte. Ainsi, les algorithmes sont classés selon la classe de graphes la plus large qu'il manipule.

Le reste du document est organisé comme suit. Dans le chapitre 1, après avoir présenté la terminologie de base utilisée dans la plupart des parties du document, nous décrirons le problème de transformation des critères subjectifs utilisés par les humains pour évaluer la qualité de l'agencement en critères exprimés sous forme de contraintes. Un catalogue de contraintes est présenté sur la base d'algorithmes qui existent. La deuxième partie du chapitre présente les différentes classes de graphes et donc d'algorithmes d'agencement. Dans la dernière partie, nous présentons la notion du noyau d'éditeur de graphes et les différentes fonctions que nous avons jugées nécessaires dans un noyau.

Le chapitre 2 présente les différents travaux sur les algorithmes d'agencement propres aux arbres binaires. Nous présentons aussi l'extension (correction) que nous avons porté à l'un des algorithmes et sa généralisation par la suite pour le cas m -air.

Le chapitre 3 décrit les différentes phases d'un algorithme d'agencement pour le cas des graphes orientés (élimination des circuits, hiérarchisation, réduction du nombre d'intersections et le calcul des coordonnées des sommets). Plusieurs heuristiques ont été étudiées et améliorées. Dans ce chapitre, nous présentons nos trois techniques pour réaliser les deux premières phases de l'algorithme.

Le chapitre 4 consacré à la classe des graphes planaires, présente deux algorithmes linéaires de test de la planarité, et une structure de données dite *PQ-arbres* utilisée pour linéariser le second algorithme.

En fin, une brève conclusion suivie de quelques perspectives sont présentées.