

Utilisation de la classe de document `these.cls`

Gilbert Babin et Mourad Sassi

Décembre 1997

1 Introduction

Dans ce document, nous voulons vous présenter la classe de document `these.cls`, développée pour la présentation de mémoires et de thèses au Département d'informatique de l'Université Laval.

Le document se veut simple et complet, illustrant chaque nouvelle commande ou option à l'aide d'un exemple.

On présentera dans l'ordre :

- les options;
- les environnements pré-définis;
- les environnements flottants pré-définis;
- les compteurs pré-définis;
- les commandes pré-définies.

2 Options

Les options sont des mots-clés passés en paramètre optionnel à la commande `\documentclass`. On peut classer ces options selon les catégories suivantes :

- Identification du format de papier utilisé On retrouve ici les options \LaTeX standards :
 - `a4paper` : Papier en format A4 (210×297 mm);
 - `a5paper` : Papier en format A5 (148×210 mm);
 - `b5paper` : Papier en format B5 (176×250 mm);
 - `letterpaper` : Papier en format lettre ($8\frac{1}{2} \times 11$ in); option par défaut;
 - `legalpaper` : Papier en format légal ($8\frac{1}{2} \times 14$ in);
 - `executivepaper` : Papier en format exécutif ($7\frac{1}{4} \times 10\frac{1}{2}$ in).

- Modification de la mise en page. On retrouve dans cette catégorie les options modifiant la mise en page :
 - `oneside` : Mise en page pour impression recto seulement; option par défaut;
 - `twoside` : Mise en page pour impression recto verso;
 - `openright` : Force le début d'une partie ou d'un chapitre sur une page de droite, lorsqu'en mode `twoside`;
 - `openany` : Force le début d'une partie ou d'un chapitre sur la prochaine page; option par défaut;
 - `onecolumn` : Mise en page sur une colonne; option par défaut;
 - `twocolumn` : Mise en page sur deux colonnes;
 - `leqno` : Mise en page des formules avec numérotation à gauche, plutôt qu'à droite;
 - `fleqn` : Mise en page des formules avec alignement à gauche, plutôt qu'au centre;
 - `draft` : Mise en page pour un brouillon;
 - `final` : Mise en page pour la version finale; option par défaut.
- Modification de la grosseur des caractères. On a les options suivantes :
 - `10pt` : Les caractères de grosseur normale sont à 10 points;
 - `11pt` : Les caractères de grosseur normale sont à 11 points;
 - `12pt` : Les caractères de grosseur normale sont à 12 points; option par défaut.
- Modification de la page titre. Ces options permettent de définir une page titre pour un mémoire (par défaut; fig. 1) ou une thèse :
 - `msc` : Une page couverture de mémoire de maîtrise est générée avec la commande `\maketitle`; option par défaut;
 - `phd` : Une page couverture de thèse de doctorat est générée avec la commande `\maketitle`.

3 Environnements pré-définis

La classe `these.cls` définit les environnements L^AT_EX standards, soient les environnements: `description`, `abstract`, `verse`, `quotation`, `quote`, `titlepage`, `thebibliography`, `theindex`.

4 Environnements flottants pré-définis

En plus des environnements flottants standards `figure`, `figure*`, `table` et `table*`, la classe `these.cls` définit deux nouveaux environnements: `algorithm`, `algorithm*`. Ces environnements permettent l'écriture d'algorithmes comme des blocs flottants. De plus, ces blocs seront automatiquement numérotés et ajoutés à la «liste des algorithmes» lorsque l'environnement `algorithm` est utilisé.

Par exemple, on pourra écrire un algorithme avec le code suivant :

```
\usepackage{algorithmic}
\begin{algorithm}
\framebox{\parbox{\textwidth}{
```

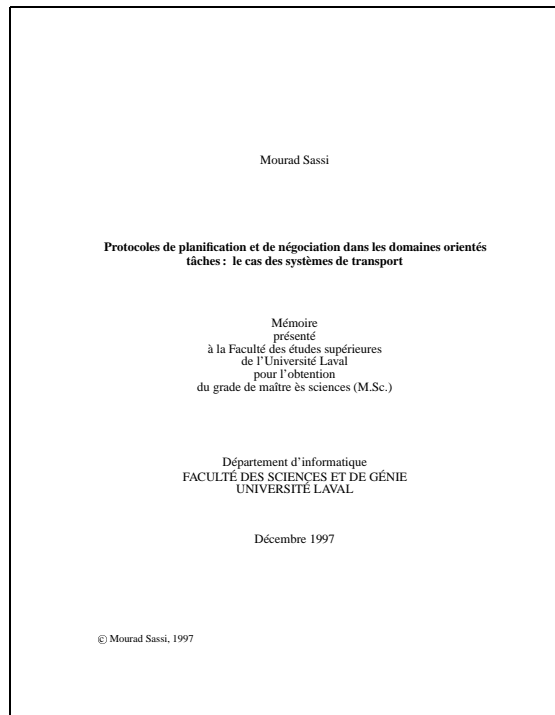


Figure 1: Page couverture par défaut

```

\begin{algorithmic}
  $IDA^*$; \\
  /* Initialisation */\\
  Soit $n_0$ la $1^{\text{ème}}$ feuille rencontrée lors
  de la visite de l'arbre en ``profondeur d'abord'';\\
  $seuil \leftarrow f(n_0)$;\\
  \REPEAT
  Développer l'arbre en ``profondeur d'abord'';\\
  \IF {$n$ est une feuille} \IF {$f(n) < seuil$} $seuil \leftarrow f(n)$
  \ENDIF
  \ELSE Élaguer la branche de l'arbre à partir de $n$;\\
  \ENDIF \UNTIL{ce qu'il ne reste qu'une branche dans l'arbre}
  Retourner cette branche;
\end{algorithmic}}
\caption{Algorithme IDA*}\label{AlgoA}
\end{algorithm}

```

Le résultat est illustré à la figure 2. On pourra aussi utiliser la commande pré-définie `\algorithme` (cf. sect. 6).

5 Compteurs pré-définis

La classe de documents `these.cls` définit les compteurs standards suivants : `part`, `chapter`, `figure`, `table`, `section`, `subsection`, `subsubsection`, `paragraph`, `subparagraph`. On notera entre autre que la numérotation des figures,

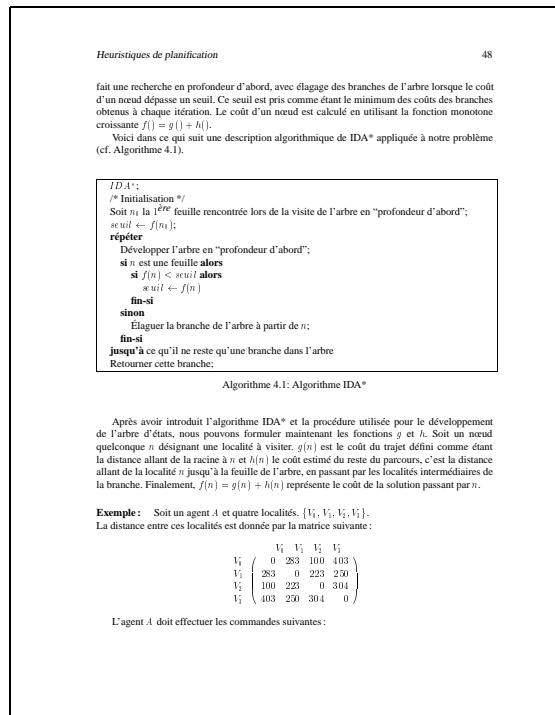


Figure 2: Écriture d'un algorithme

tables et sections est remise à zéro dans un nouveau chapitre.

Nous avons aussi défini un compteur pour l'environnement `algorithm`, soit le compteur du même nom (`algorithm`). Ce compteur est réinitialisé à zéro d'un chapitre à l'autre.

6 Commandes pré-définies

Comme toute classe qui se respecte, `these.cls` définit la grande majorité des commandes standards de \LaTeX . Il faut aussi savoir que la classe a été construite à partir de la classe `report.cls`. Les informations sur la classe `report.cls` qu'on retrouve dans la documentation \LaTeX s'appliquent pour la plupart à la classe `these.cls`. On retrouve donc les commandes: `\maketitle`, `\thepart`, `\thechapter`, `\thesection`, `\thesubsection`, `\thesubsubsection`, `\theparagraph`, `\thesubparagraph`, `\part`, `\chapter`, `\section`, `\subsection`, `\subsubsection`, `\paragraph`, `\subparagraph`, `\theenumi`, `\theenumii`, `\theenumiii`, `\theenumiv`, `\labelenumi`, `\labelenumii`, `\labelenumiii`, `\labelenumiv`, `\labelitemi`, `\labelitemii`, `\labelitemiii`, `\labelitemiv`, `\descriptionlabel`, `\appendix`, `\theequation`, `\thefigure`, `\thetable`, `\tableofcontents` (fig. 3 et 4), `\listoffigures` (fig. 5), `\listoftables` (fig. 6), `\contentsname`, `\listfigurename`, `\listtablename`, `\bibname`, `\indexname`, `\figurename`, `\tablename`, `\partname`, `\chaptername`, `\appendixname`, `\abstractname`, `\today`.

De plus, on a défini quelques commandes utilitaires :

- `\annee` : Cette commande permet de définir l'année d'écriture du mémoire. On l'utilise en conjonction avec les com-

Table des matières	
Résumé	1
Avant-propos	2
Table des matières	3
Liste des tableaux	6
Liste des algorithmes	7
Liste des figures	8
1 Introduction générale	1
1.1 Motivations	3
1.2 Objectifs	3
1.3 Organisation du mémoire	4
2 Négociation dans le domaine orienté tâche	5
2.1 Le problème de la négociation	5
2.1.1 Définition de la négociation	5
2.1.2 Les attributs de la négociation	6
2.1.3 La fonction "utilité"	7
2.1.4 Les différents domaines de la négociation	7
2.1.5 Mécanisme de négociation	10
2.2 Ententes dans les DOT : application au transport	11
2.2.1 Classification du domaine du transport	11
2.2.2 Définition du domaine du transport	11
2.2.3 Caractérisation du domaine	12
2.2.4 Rencontre dans un DOT	14
2.2.5 Entente dans un DOT	15
2.2.6 Entente conflictuelle	15
2.2.7 Utilité dans un DOT	16
2.2.8 Catégorisation du domaine orienté tâche	17
2.3 Protocoles de négociation	19
2.3.1 Le protocole monotone de concession	20

Figure 3: Table des matières

Table des matières	4
2.3.2 Le protocole à une étape	20
2.4 Stratégies de négociation	21
2.4.1 La stratégie de Zeuthen	21
2.4.2 Évaluation du risque	22
2.5 Évaluation du mécanisme de négociation	22
2.5.1 Efficacité du mécanisme	22
2.5.2 Stabilité du mécanisme	23
2.5.3 Simplicité du mécanisme	25
2.6 Autres mécanismes de négociation	25
2.6.1 Protocole de tromperie	25
2.6.2 Mécanismes maximisant le produit des utilités	25
2.6.3 Mécanisme basé sur le "Contract Net"	26
2.6.4 Mécanisme basé sur le commerce simulé	26
2.7 Conclusion	26
3 Protocoles et stratégies de négociation dans le domaine du transport	27
3.1 Stratégie générale de négociation	27
3.2 Le protocole "Contract Net"	29
3.3 Protocoles de décomposition de tâches	31
3.3.1 Définition de la décomposition de tâche	31
3.3.2 Modèle centralisé de décomposition de tâches	31
3.3.3 Modèle décentralisé de décomposition de tâches	33
3.3.4 Modèle complètement décentralisé de décomposition de tâches	34
3.4 Le protocole basé sur le commerce simulé	35
3.4.1 Exemple	35
3.4.2 Description de la fonction coût	36
3.4.3 Description de l'algorithme du commerce simulé	37
3.4.4 Phase d'achat et de vente de tâches	38
3.4.5 Phase de recherche de relations d'appariement	39
3.5 Exemple de négociation	40
3.6 Évaluation du mécanisme de négociation basé sur le commerce simulé	41
3.6.1 Efficacité du mécanisme	41
3.6.2 Stabilité du mécanisme	41
3.6.3 Simplicité du mécanisme	41
3.6.4 Distribution du mécanisme	42
3.6.5 Symétrie du mécanisme	42
3.7 Conclusion	42
4 Heuristiques de planification	43
4.1 Description des schémas de résolution choisis	43
4.1.1 Description des fonctions	45
4.2 L'algorithme A*	45
4.2.1 Espace d'états dans le domaine du transport	46
4.2.2 Application au domaine du transport	47

Figure 4: Table des matières (suite)

Liste des figures

2.1	Catégorisation des domaines	7
2.2	Négociation dans un DOT	9
2.3	Négociation dans un DOÉ	10
2.4	Organisation du DOT dans le domaine du transport	13
2.5	Une rencontre dans le domaine des livraisons	14
2.6	Le domaine du transport n'est pas globalement subadditif	19
2.7	Le protocole monotone de concession	20
2.8	Le facteur de risque	21
2.9	Rencontre dans le domaine des livraisons	23
3.1	Organisation hiérarchique du système	28
3.2	Description générale du système	29
3.3	Le modèle centralisé de décomposition de tâches	32
3.4	Négociation centralisée dans le domaine des transports	32
3.5	MDDT dans le domaine des transports	34
3.6	Graphe commercial	36
3.7	La suppression ou l'insertion dépendent de la fonction coût	37
3.8	Exemple de négociation entre agents	40
4.1	Description du protocole "Contract Net"	44
4.2	Recherche du plan de coût minimal dans un arbre	47
4.3	Arbre de recherche	49
4.4	Recherche d'un état à coût minimal avec le recuit thermique	51
4.5	Une simple hybridation	56
4.6	Une simple mutation	57
4.7	L'opérateur d'hybridation	59
4.8	L'opérateur de mutation	59
5.1	Apport du CS sur le CNP	63
5.2	Performances des différentes stratégies	63
5.3	Comparaison du CS et de l'algo. génétique	64
5.4	Vitesse de convergence de la stratégie C.S. + R.S. en fonction du temps	65
5.5	Vitesse de convergence de la stratégie C.S. + Tabou en fonction du temps	65
5.6	Comparaison des différentes stratégies en fonction du temps d'exécution	66
5.7	Comparaison des différentes stratégies en fonction des capacités	67

Figure 5: Liste des figures

Liste des tableaux

2.1	Ententes dans le domaine des livraisons	17
2.2	L'ensemble de négociation dans le domaine des livraisons	23
2.3	La dernière étape dans la forme normale de jeu	24
2.4	Le jeu du poulet	25
4.1	Résultats après une génération pour $f(x) = x^2$	57
5.1	Résultats expérimentaux - IDA*, recuit, tabou, génétique	62

Figure 6: Liste des tableaux

mandes `\title`, `\author` et `\date` pour construire la page couverture avec `\maketitle`. L'exemple de la figure 1 a été construit avec les commandes suivantes :

```
\author{Mourad Sassi}
\title{Protocoles de planification et de négociation dans
       les domaines orientés tâches : le cas des
       systèmes de transport}
\date{Décembre 1997}
\annee{1997}
\maketitle
```

- `\thesisbody` : Cette commande réinitialise la pagination une fois que les sections en préface sont terminées (résumé, avant-propos, table des matières, etc.). On place cette commande juste avant le premier chapitre. En fait, cette commande est similaire à la commande `\appendix` qui modifie la mise en page pour les annexes. Par exemple, on pourrait avoir les commandes suivantes :

```
\include{resume}
\include{av-propos}

\tableofcontents
\listoftables
\listofalgorithms
\listoffigures

\thesisbody

\include{introduction}
\include{chap2}
\include{chap3}
\include{chap4}
\include{chap5}
\include{conclusion}

\bibliographystyle{babinfr}
\bibliography{biblio}

\appendix

\include{annexea}
\include{annexeb}
```

- `\thealgorithm` : Cette commande imprime un numéro d'algorithme (numéro de chapitre et numéro d'algorithme dans le chapitre).
- `\listofalgorithms` : Cette commande produit la liste des algorithmes (fig. 7).
- `\listalgorithmname` : Le nom de la liste des algorithmes imprimé dans le document.
- `\algorithmname` : Le nom utilisé dans la légende d'un algorithme.
- `\algoinsert` : Fonction utilitaire permettant l'insertion d'un algorithme dans un cadre (fig. 2). La commande utilise 5 paramètres :

1. La position de l'algorithme dans le texte (h pour «here», t pour «top», b pour «bottom», p pour «page of floats»). Ce paramètre est optionel et prends la valeur ht par défaut;

Liste des algorithmes	
3.1	Algorithme du "Contract Net" 30
3.2	Description générale de l'algorithme du commerce simulé 37
3.3	Procédure d'achat/vente et de recherche de relations d'appariement 38
4.1	Algorithme IDA* 48
4.2	Algorithme du recuit simulé 50
4.3	Algorithme de la recherche tabou 54
4.4	Algorithme génétique "simple" 55
4.5	Algorithme génétique 58

Figure 7: Liste des algorithmes

2. La largeur de l'algorithme dans la page;
3. La légende de l'algorithme;
4. L'étiquette utilisée pour référer à cet algorithme;
5. L'algorithme lui-même.

L'exemple d'utilisation de l'environnement `algorithm` peut être modifié en utilisant `\algorithinsert` comme suit :

```
\usepackage{algorithmic}
\algorithinsert{\textwidth}{Algorithme IDA*}{AlgoA}{%
\begin{algorithmic}
...
\end{algorithmic}%
}
```

- `\figinsert`: Fonction permettant l'insertion d'une figure (fig. 8). La commande utilise 4 paramètres :
 1. La position de la figure dans le texte (h pour «here», t pour «top», b pour «bottom», p pour «page of floats»). Ce paramètre est optionel et prends la valeur ht par défaut;
 2. La légende de la figure;
 3. L'étiquette utilisée pour référer à cette figure;
 4. La figure elle-même.

On peut insérer une figure avec la commande suivante :

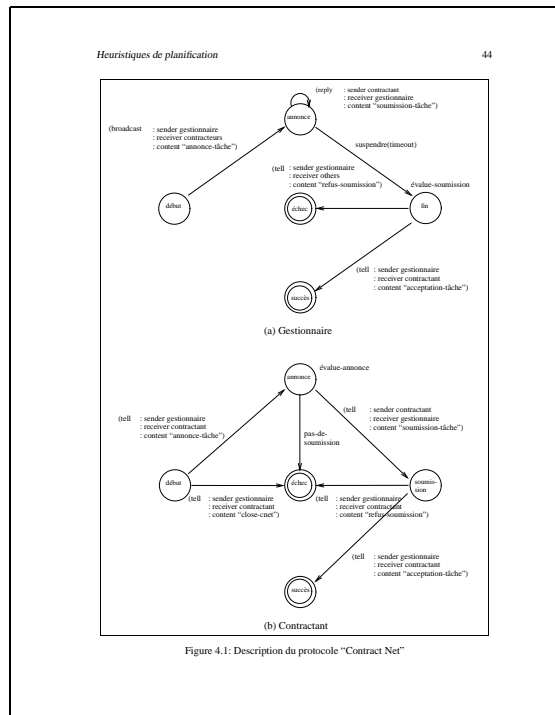


Figure 8: Utilisation de `\figinsert`

```
\figinsert{Description du protocole ``Contract Net''}{cnpmc2}{%
...
}
```

- `\epsfiginsert`: Fonction utilitaire permettant l'insertion d'un fichier EPS comme une figure (fig. 9). La commande utilise 5 paramètres :
 1. La position de la figure dans le texte (h pour «here», t pour «top», b pour «bottom», p pour «page of floats»). Ce paramètre est optionnel et prends la valeur ht par défaut;
 2. Le nom du fichier EPS contenant la figure;
 3. La largeur relative de la figure par rapport à la largeur de la ligne courante; Une valeur de 1 indique que la figure prend la largeur de la ligne, alors qu'une largeur de 0.5 indique que la figure prend la moitié de la largeur;
 4. La légende de la figure;
 5. L'étiquette utilisée pour référer à cette figure.

On peut insérer un fichier EPS comme figure avec la commande suivante :

```
\epsfiginsert[htb]{crossover.eps}{.7}{L'opérateur d'hybridation}
{crossover-rbx}
}
```

B. La sélection : Durant cette phase, les solutions parentes sont sélectionnées à partir de la population locale. Le processus de sélection est stochastique et biaisé envers les meilleures solutions. La valeur d'adaptation nous donne une indication sur la qualité de la solution. Une solution A est meilleure qu'une solution B si $distance(A) < distance(B)$.

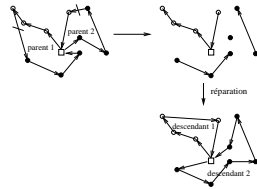


Figure 4.7: L'opérateur d'hybridation

C. L'hybridation : Cet opérateur crée un descendant en remplaçant une portion de plan de la solution parent 1 par une portion de plan de la solution parent 2. Cette phase peut laisser certaines localités du plan isolées ou dupliquées. Dans la figure 4.7, le plan avec les localités noires remplace le plan correspondant dans la solution parent 2. On remarque qu'il y a des localités qui appartiennent à la fois à deux plans, tandis que certaines localités restent isolées. Un opérateur de réparation est appliqué aux descendants dans le but de générer un plan faisable.

D. La mutation : L'opérateur de mutation sélectionne un plan et déplace une localité à visiter à une position aléatoire du plan d'un agent (cf. Figure 4.8).



Figure 4.8: L'opérateur de mutation

Figure 9: Utilisation de `\epsfiginsert`