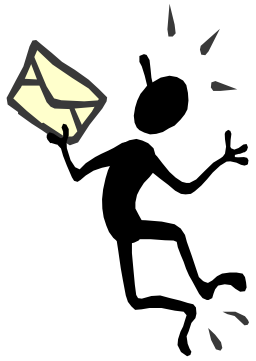
Benchmark et automatisation du tuning des algorithmes de matching de schémas XML

Encadrement : Nabila BENHARKAT
Rami RIFAIEH

Présenté par : Mohamed BOUKHEBOUZE



Atelier *DECOR*

Contexte

- Les données des différents systèmes d'information sont hétérogènes.
- Les utilisateurs ont besoin d'une vue intégrée.

→ Le *matching* de schémas

Contexte

- « *Le matching de schéma est une tâche qui consiste à trouver les correspondances sémantiques entres les éléments de deux schémas* » [Do et Rahm 2002].

- Domaines d'application :
  - Web sémantique,
  - L'intégration de schémas et d'ontologies,
  - L'e-commerce.

Contexte

- Le *matching* a été effectué jusque là manuellement à l'aide d'une interface graphique

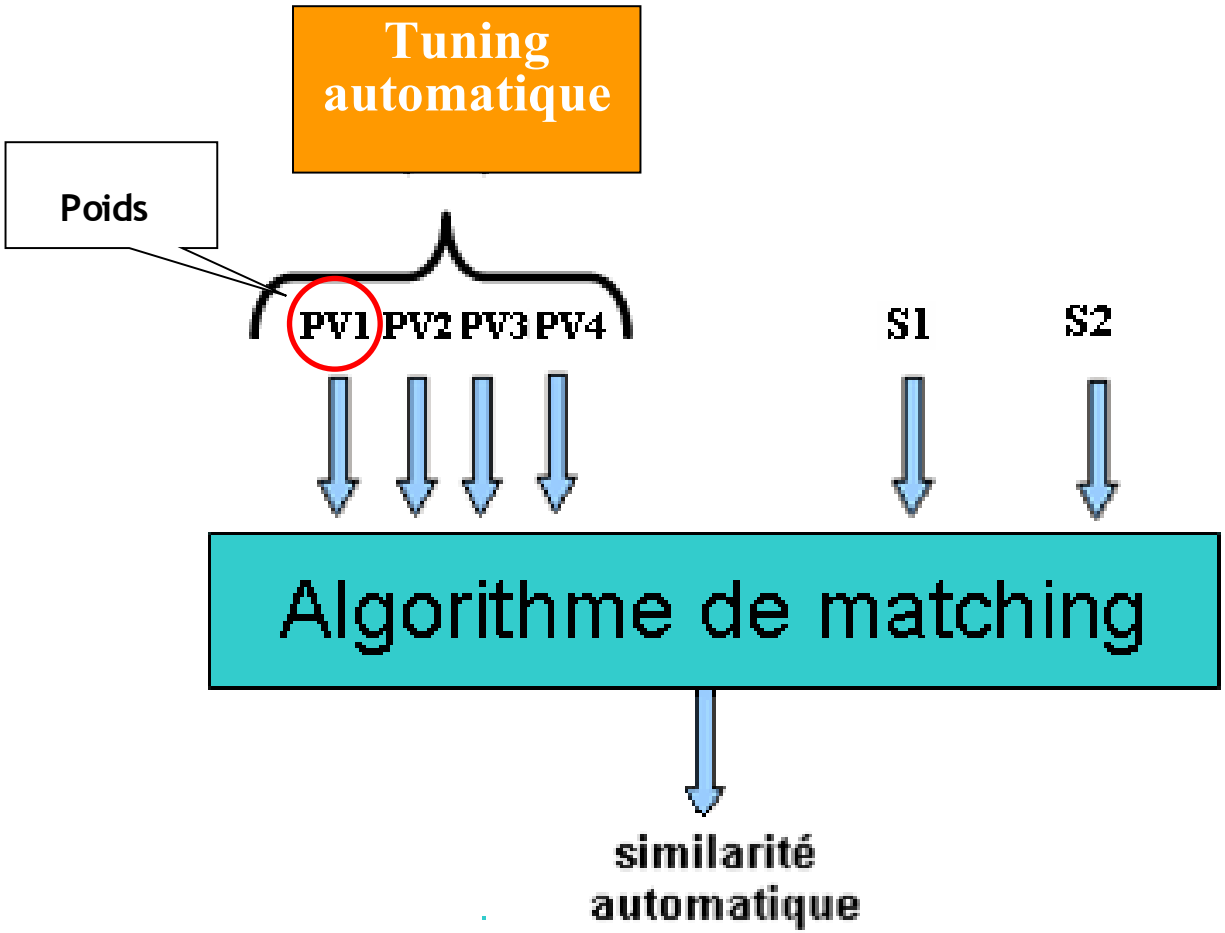
Problème

- Consommation du temps
- Intervention des experts humains

→ Automatiser le processus de *matching*

- Plusieurs algorithmes ont été proposés : COMA [*Rahm 02*], EXSMAL [Chukmol &al *04*], SCIA [*Wang 04*], Cupid [*Madhavan 01*], SF [*Sergey 01*] ET Xmapper [*Kurgan 02*]

Contexte



Objectifs

- Déterminer les meilleures valeurs de paramètres qui servent à calculer la similarité structurelle.
- Proposer un benchmark qui examine comment les différents paramètres structurels des algorithmes influent sur les résultats.
- Proposer un algorithme pour automatiser le choix des paramètres structurels optimisés en se basant sur les relations déduites à partir du benchmark.

Benchmark

- Un benchmark est utile pour comparer l'efficacité de différents systèmes matching

→ Très peu de benchmarks de matching proposés

- Des évaluations sur les algorithmes de matching qui sont toujours individuelles

Évaluation de COMA

- Cinq schémas XML avec des caractéristiques différentes
- Dix tâches de matching sont définies, pour chaque tâche deux différents schémas
- 12312 séries de tests pour chaque stratégie on com
- Une
- Avec d prédéfinis. La précision, le Recall et le Overall sont calculés

indique le nomb correspondanc parmi celles retourn

indique le nombre de vrais corresp parmi cor

$$Recall * (2 - \frac{1}{precision})$$

Tuning automatique

- Le tuning est la tâche qui consiste à ajuster les paramètres des différents matchers

→ Il existe peu de travaux proposés pour tuning automatique :

- Berkovsky 2005 : les algorithmes génétiques pour le tuning automatique.
- Sayyadian 2005 : un système complètement automatique (eTuner)

Les algorithmes génétiques pour le tuning automatique

+ L'utilisation des AG permet d'avoir une solution générique au problème de tuning automatique

- L'efficacité des algorithmes génétiques dépend du choix de la fonction objective

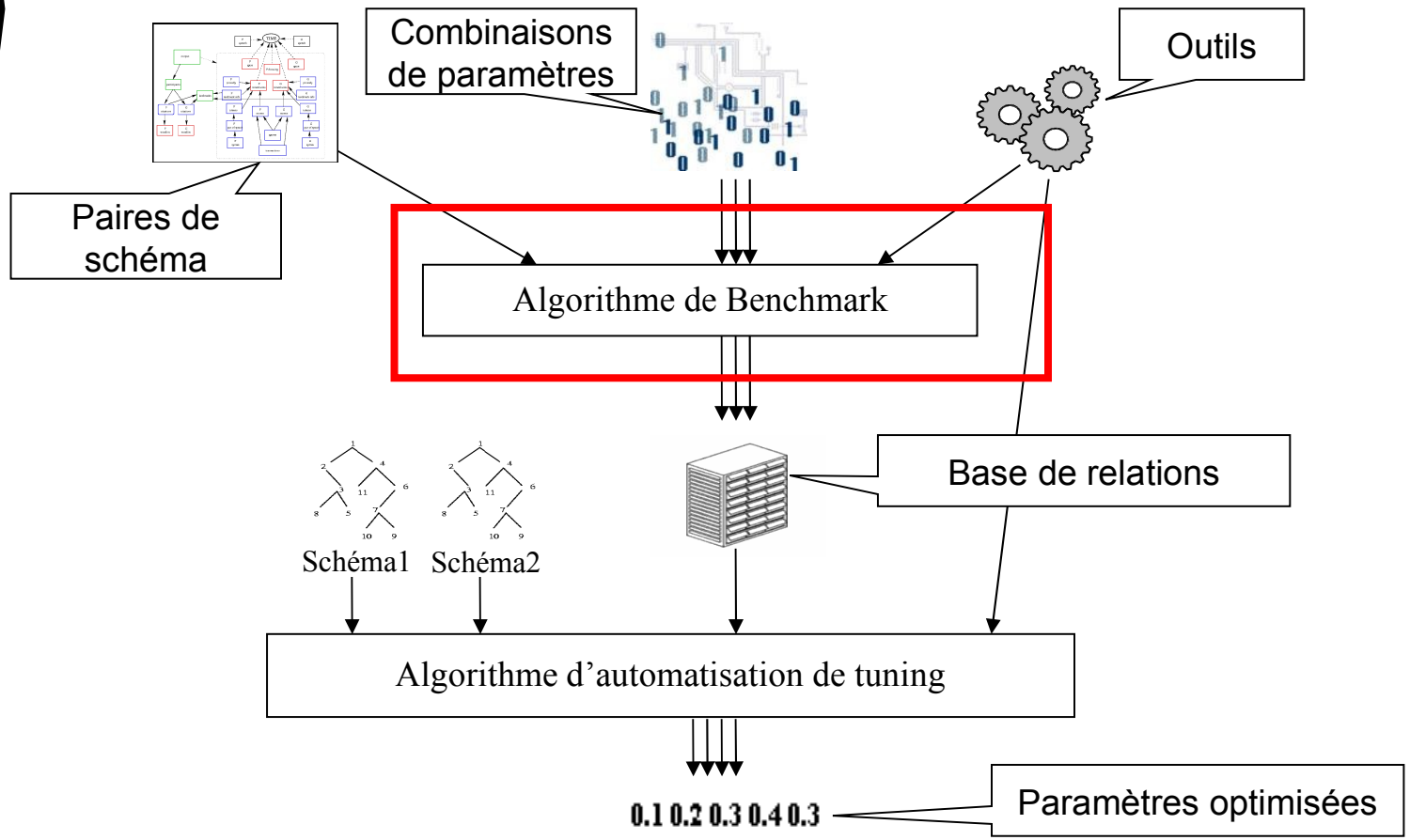
eTuner

+ Le eTuner a de bonnes performances

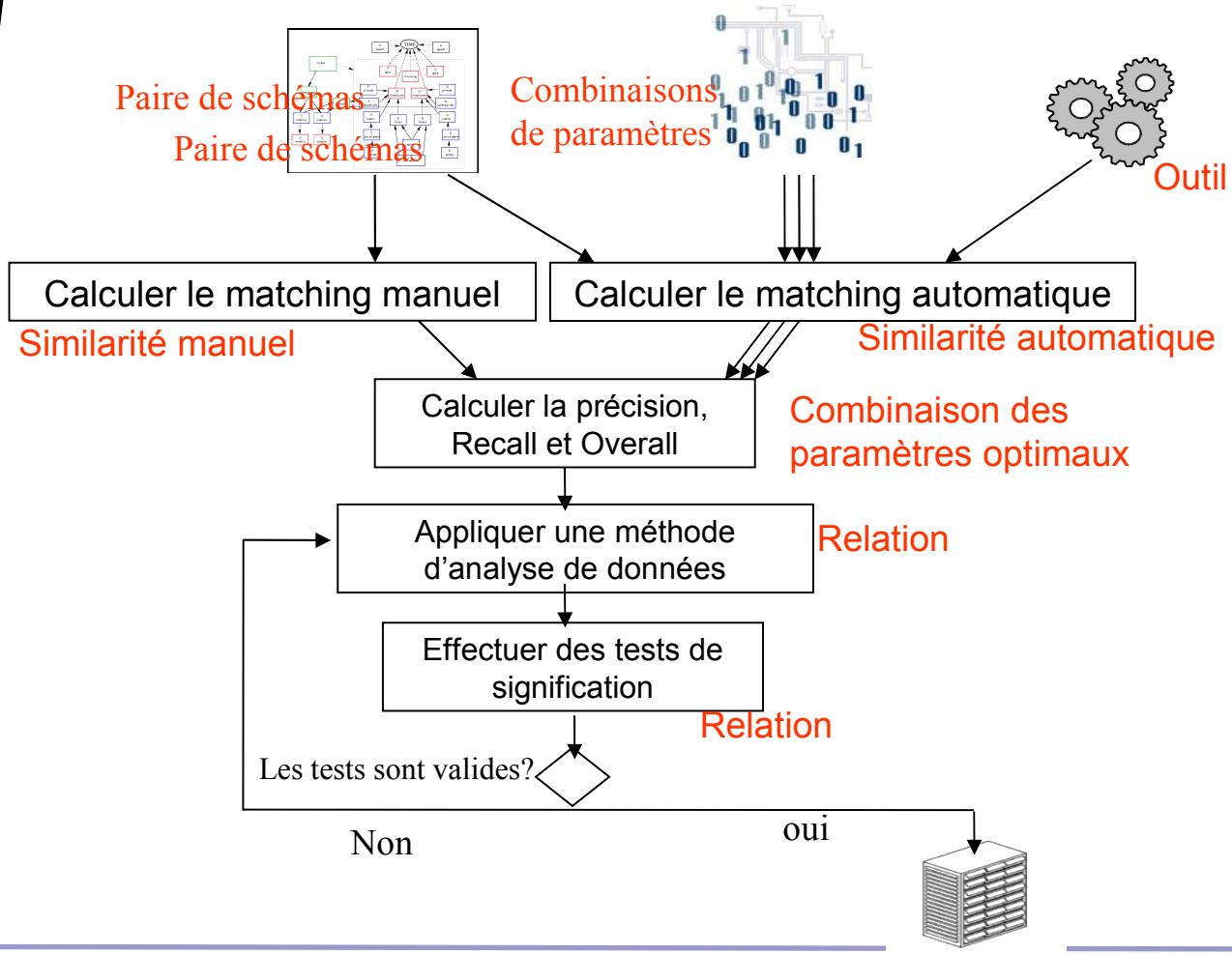
- Passe par deux étapes
  - Perturbation des schémas
  - Exécution du tuning

→ coût considérable en temps.

Notre approche



Étape de benchmark



Étape de benchmark
Schémas de tests

- **20 schémas différents utilisés :**

  N,P ∈]0,80] et D ∈ [2,7]

Où **N : Nombre d'éléments**

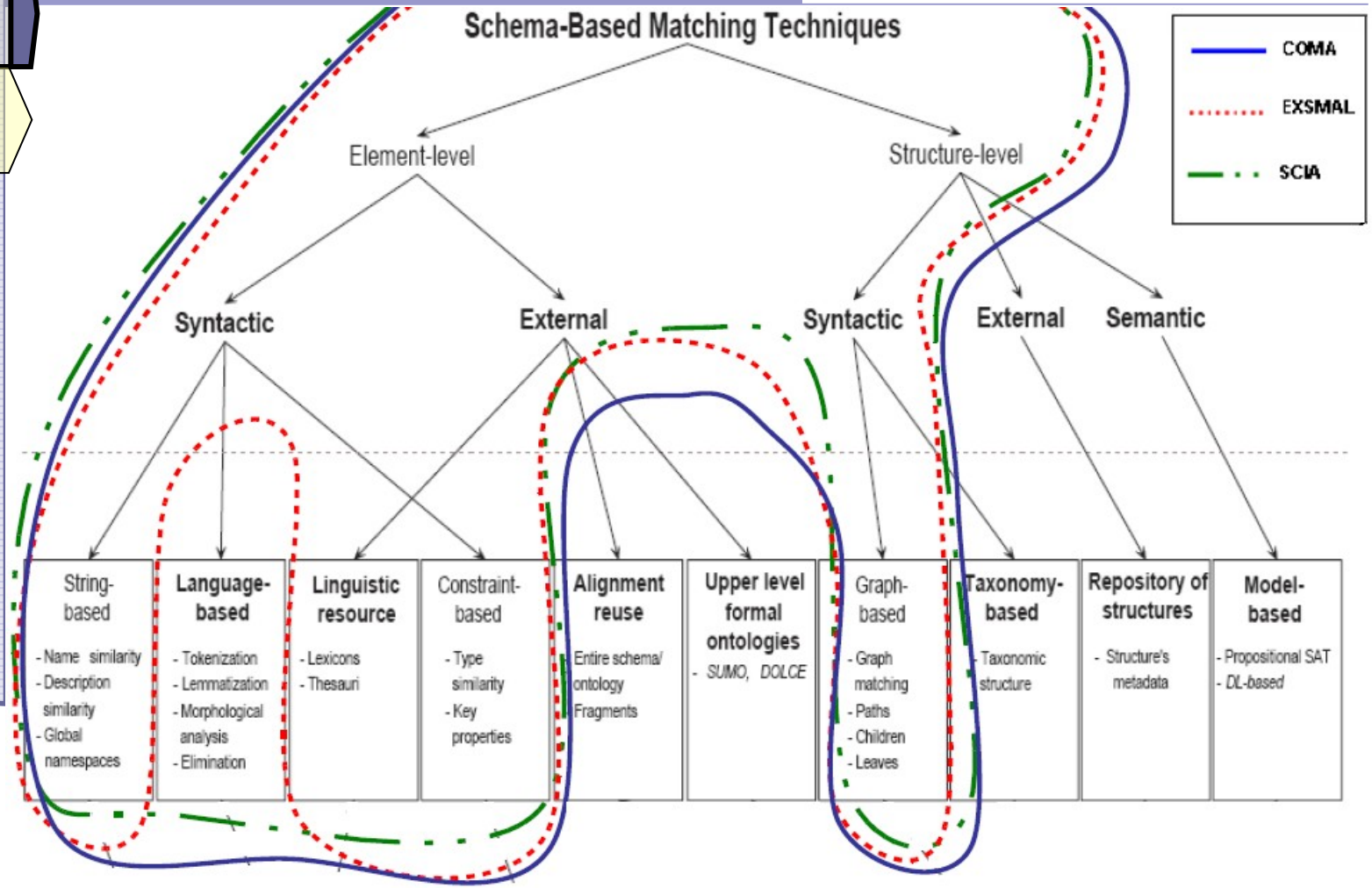
P : Nombre de chemins

D : La profondeur

→ C_{20}^{2} **= 190 combinaisons de paires**

de schémas possibles

Étape de benchmark
Algorithme de matching choisis



Étape de benchmark
Paramètres structurels étudiés

- EXSMAL : 4 Paramètres
  combinaisons possible → 20
- COMA : 4 Paramètres
  combinaisons possible → 625
- *SCIA :* 3 Paramètres
  combinaisons possible → 125

Étape de benchmark
Nombres de séries de tests

- EXSMAL : 190 X 20 = 3800
- COMA [illegible] 118
- *SCIA :* [illegible] 237

Nombre de combinaisons de paires de schémas

Nombre de combinaisons de paramètres

→ 146300

Nombre de séries de tests

Algorithme de benchmark

Pour chaque(Oi ∈ ***EO) faire*** *//(nous avons choisi 3 outils :EXSMAL , COMA et SCIA)*

Pour chaque *((Si,Sj)* ∈ *ES)* ***faire*** *// (190 paire de schémas possibles)*

Pour chaque *(combinaison de paramètres* ∈ ***EP) faire***

// (les combinaisons possibles de paramètres avec une incrémentation delta)

// Calculer le matching avec la paire de schémas;

//Calculer la précision, le Recall et le Overall;

(PR, RE, **OV**) Matching (Si, Sj, EC1);

Fin_Pour

//Trouver le modèle de régression

//Appliquer les tests de signification

Si *les tests sont valides* ***alors*** *Sauvegarder le modèle*

Sinon *Appliquer d'autres méthodes d'analyse de données*

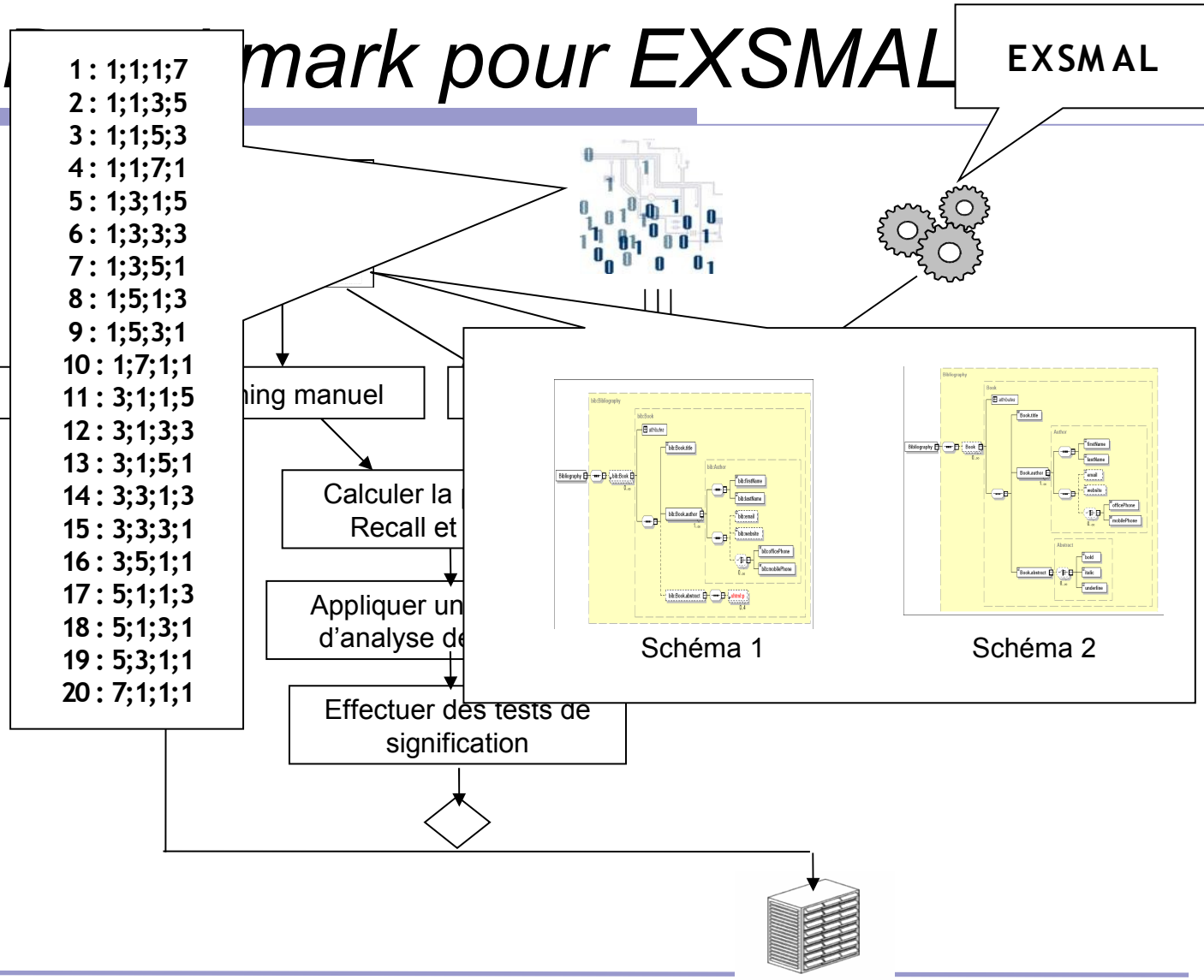
Fin_Si

Fin_Pour

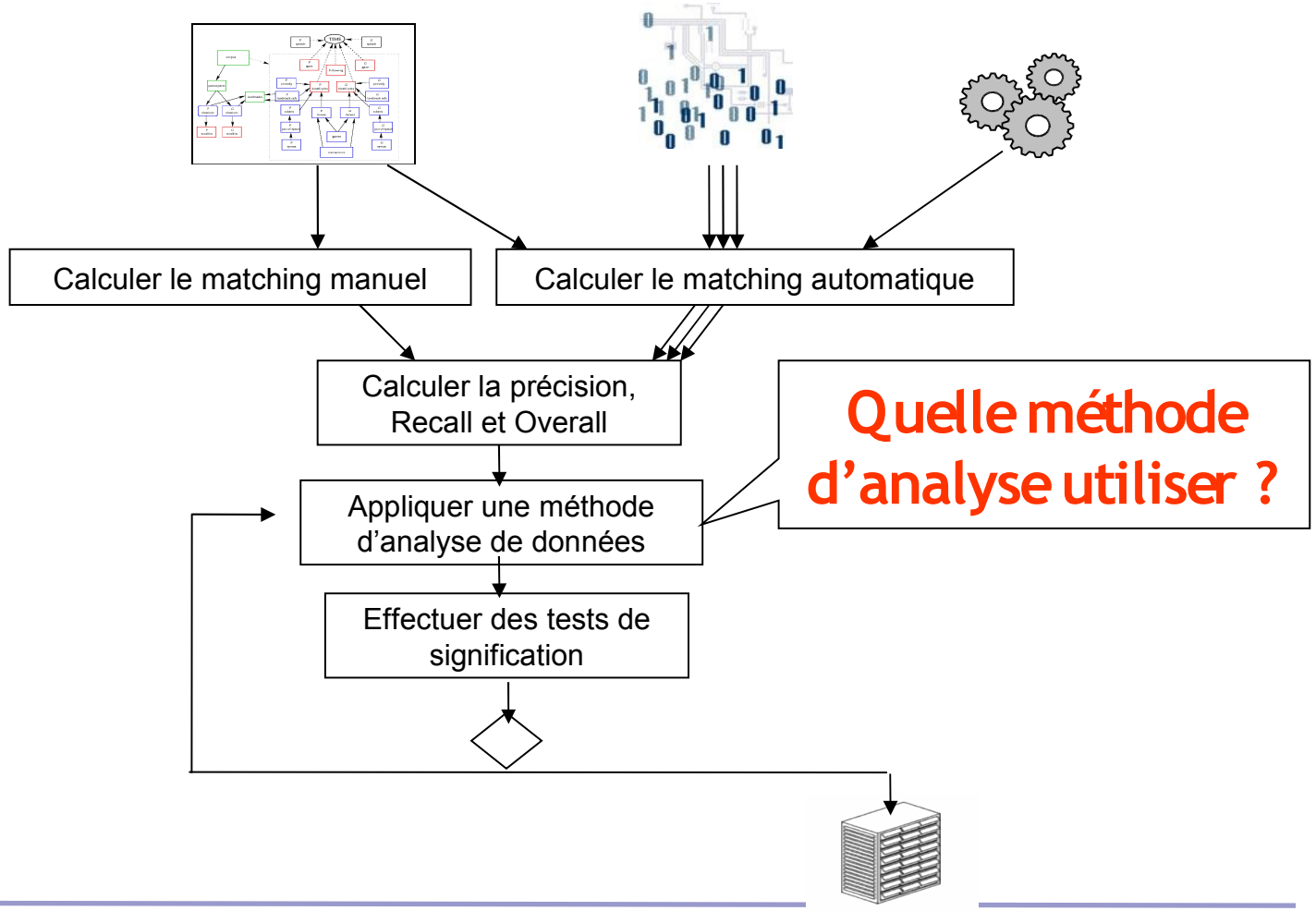
Fin_Pour

Étape de benchmark

mark pour EXSMAL



Étape de benchmark
Benchmark pour EXSMAL



Étape de benchmark
Analyse de données

- **Quelle méthode d'analyse de données utilisée?**
- **Les méthode[illegible]s :**
  - **Les ACP**
  - **La régression multiple**

Réduire le nombre de variables à étudier

Calculer l'équation de l'hyperplan

$$Y = a_0 + a_1 x_1 + a_2 x_2 + e$$

variable expl[illegible]

variable explicative

Erreur de mesure

comment juger que la relation est significative ?

Étape de benchmark
Tests de significations

- Analyse des résidus
- Le coefficient de détermination
- Les tests d'hypothèses (tes

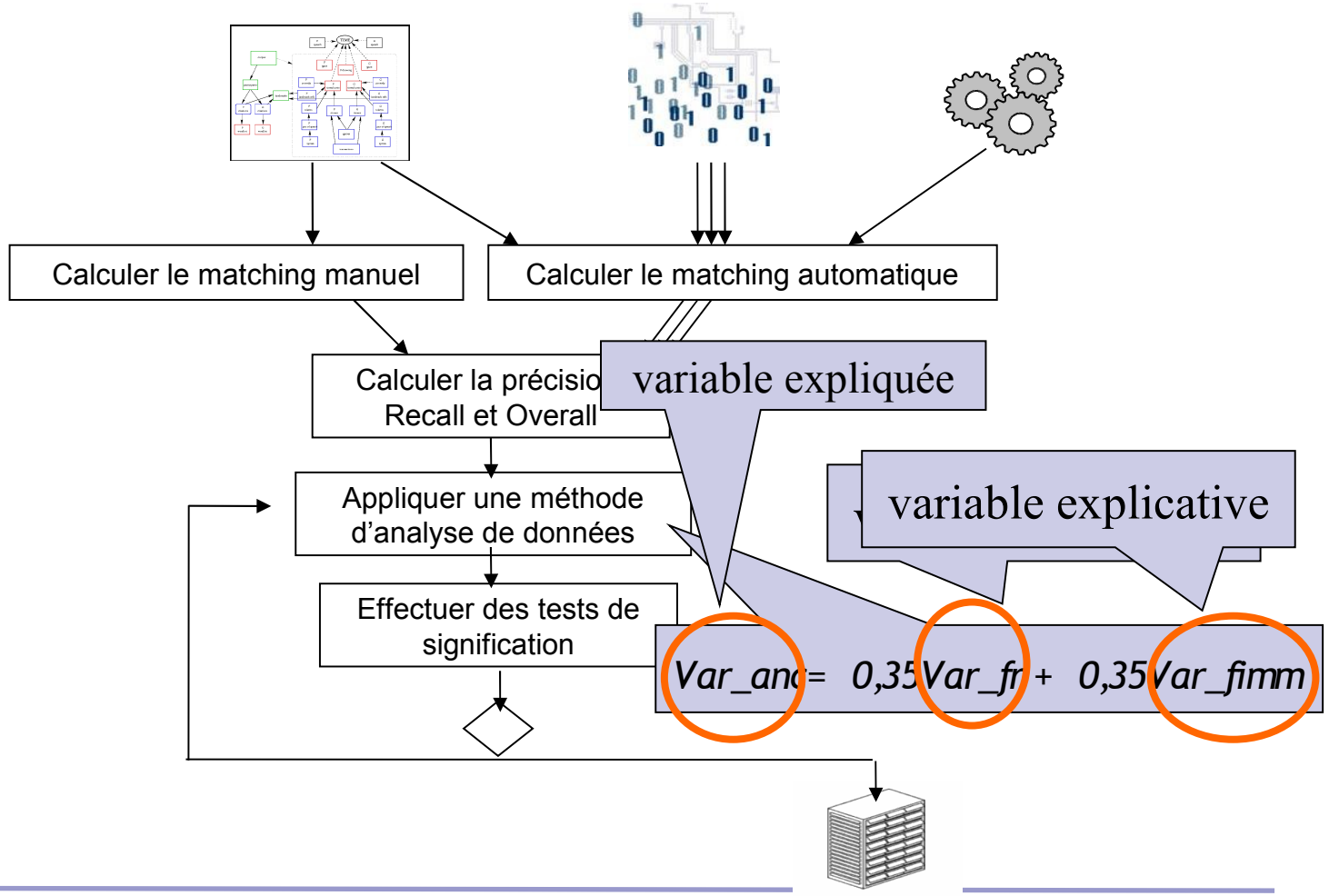
Les erreurs de mesure

La qualité de la relation

Accepter ou rejeter une hypothèse

[Gérald Baillargeon, Livre : Méthodes statistiques, les éditions SMG, 2004 p.273-40

Étape de benchmark
Benchmark pour EXSMAL



Benchmark pour EXSMAL

- coefficient de détermination $R^2 = 0,79$.
- Residus normalisés
- L'hypothèse statisti

H0: le *coeff_anc* est linéairement independant des *coeff_fr* et *coeff_fimm.*

D'après le test de Fisher

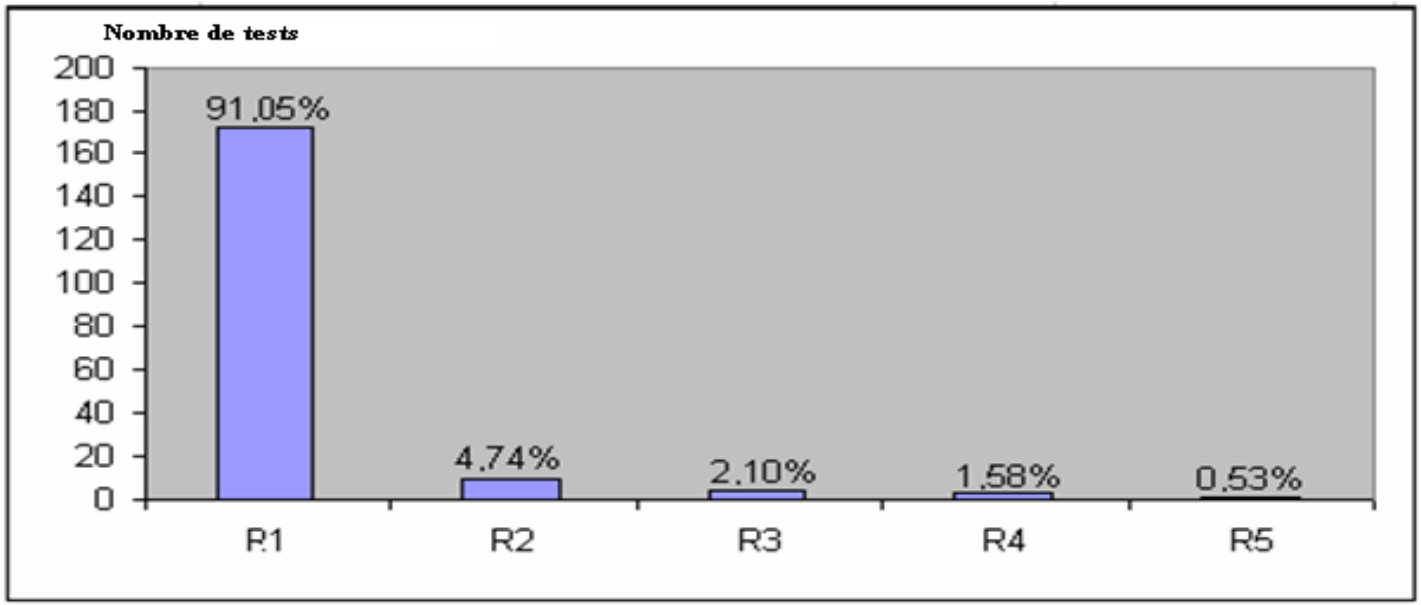
Bonne qualité

Erreurs acceptables

Hypothèses rejetées

Benchmark pour EXSMAL

Relation	La relation mathématique	N et P	D	Nombre de tests
R1	Coeff_anc = 0.35Coeff_fr + 0.35Coeff_fimm	]0,80]	[2,7]	173
R2	Coeff_anc = 0.33Coeff_fr + 0.33Coeff_fimm	]0,80]	[2,7]	9
R3	Coeff_anc = 0.25Coeff_fr + 0.04Coeff_fimm	]0,80]	[2,7]	4
R4	Coeff_anc = 0.13Coeff_fr + 0.27Coeff_fimm	]0,80]	[2,7]	3
R5	Coeff_anc =2.2Coeff_fr + 0.45Coeff_fimm	]0,80]	[2,7]	1



Benchmark pour COMA

- *P_ Parents = a0 + a1P_Children + a2P_Leaves*
- *P_ Siblings* = 10

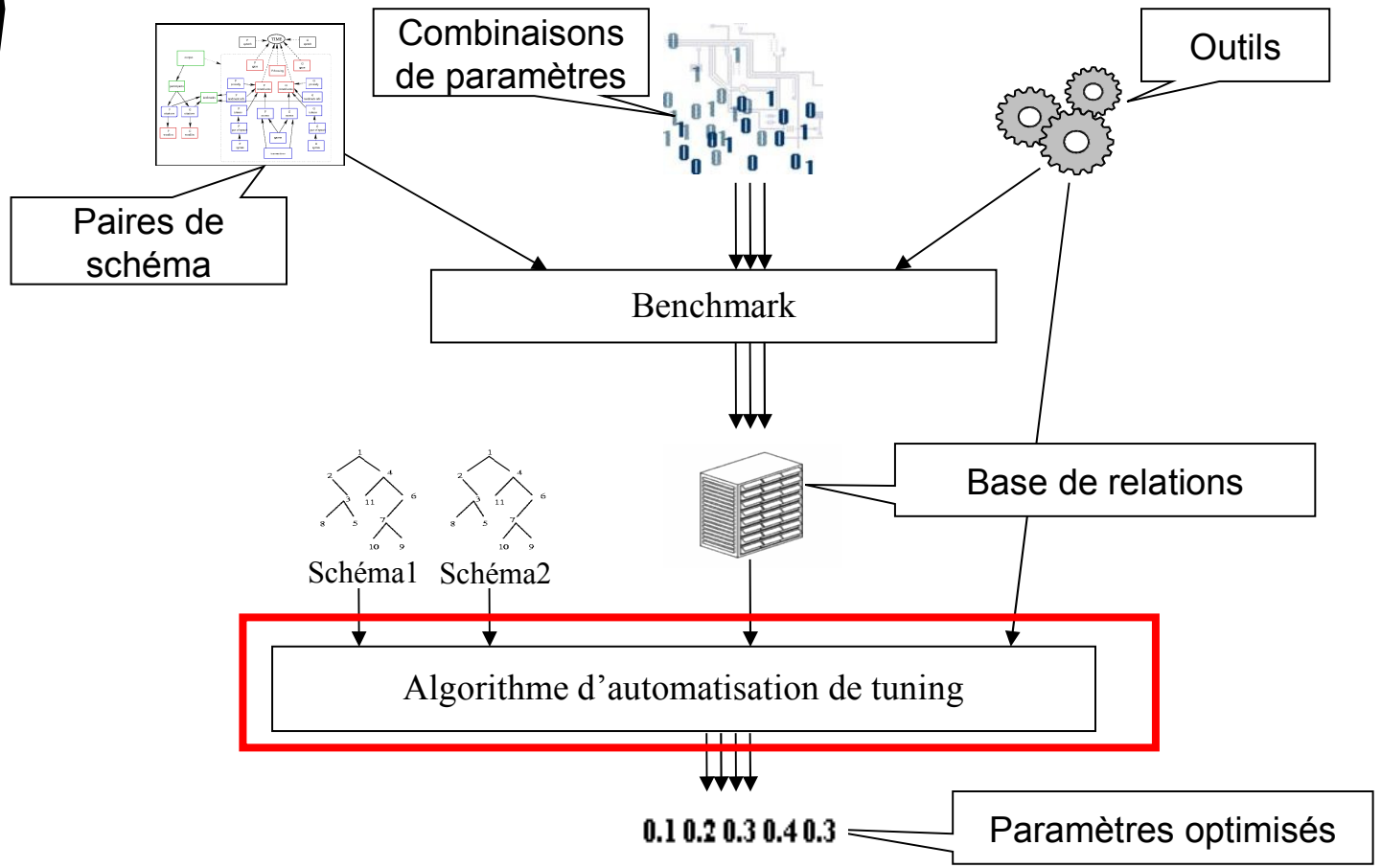
Relation	La relation mathématique	N et P	D	Nombre de tests
R1	P_ Parents = 0,6 P_Children + 0,6 P_Leaves	]0,40[	[2,4]	94
R2	P_ Parents = 0,48 P_Children + 0,48 P_Leaves	[40,80]	[2,4]	27
R3	P_ Parents = 0,11P_Children + 0,11 P_Leaves	]0,40[	[5,7]	51
R4	P_ Parents = 0,07 P_Children + 0,07 P_Leaves	[40,80]	[5,7]	18

Benchmark pour SCIA

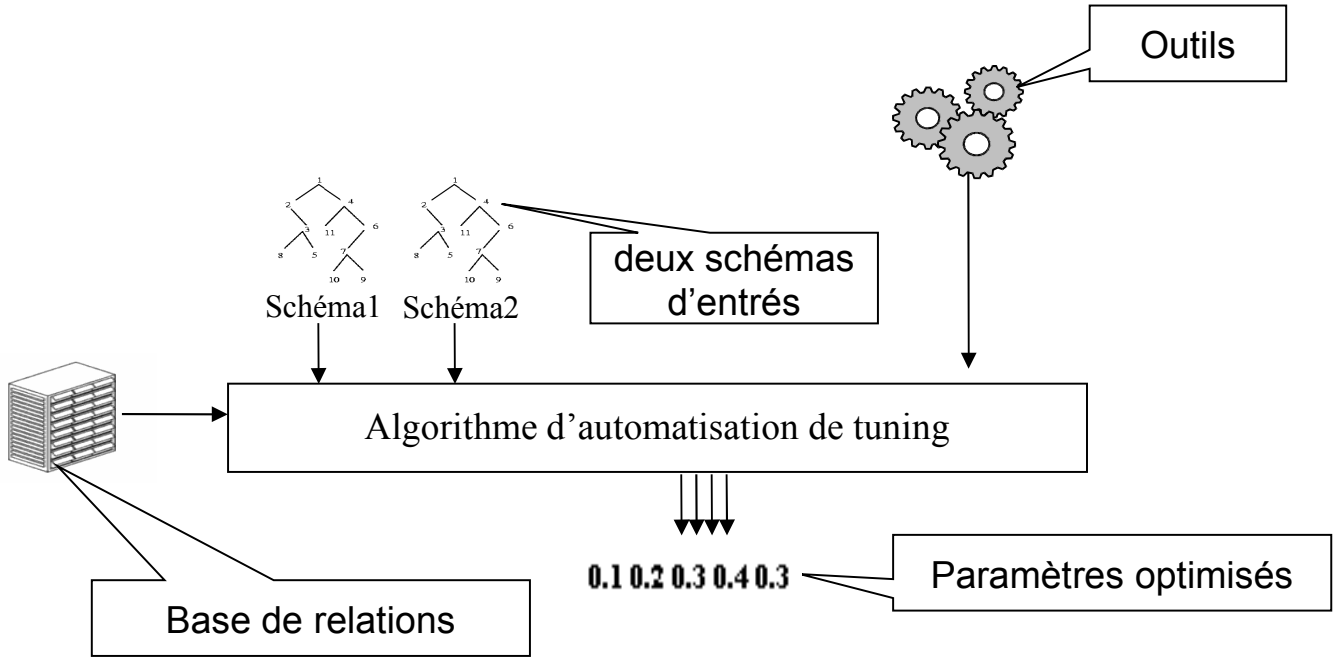
- *PATH_WEIGHT = a0 + a1GRAPH_WEIGHT + a2STRUCT_WEIGHT*

Relation	La relation mathématique	N et P	D	nombre de test
R1	PATH_WEIGHT = -0,23 RAPH_WEIGHT - 0,62 STRUCT_WEIGHT+0,51	]0,40[	[2,4]	94
R2	PATH_WEIGHT = -0,5 RAPH_WEIGHT - 0,5 STRUCT_WEIGHT+0,5	[40,80]	[2,4]	27
R3	PATH_WEIGHT = -0,25 RAPH_WEIGHT - 0,75 STRUCT_WEIGHT+0,55	]0,40[	[5,7]	51
R4	PATH_WEIGHT = -0,49 RAPH_WEIGHT - 0,33 STRUCT_WEIGHT+0,52	[40,80]	[5,7]	18

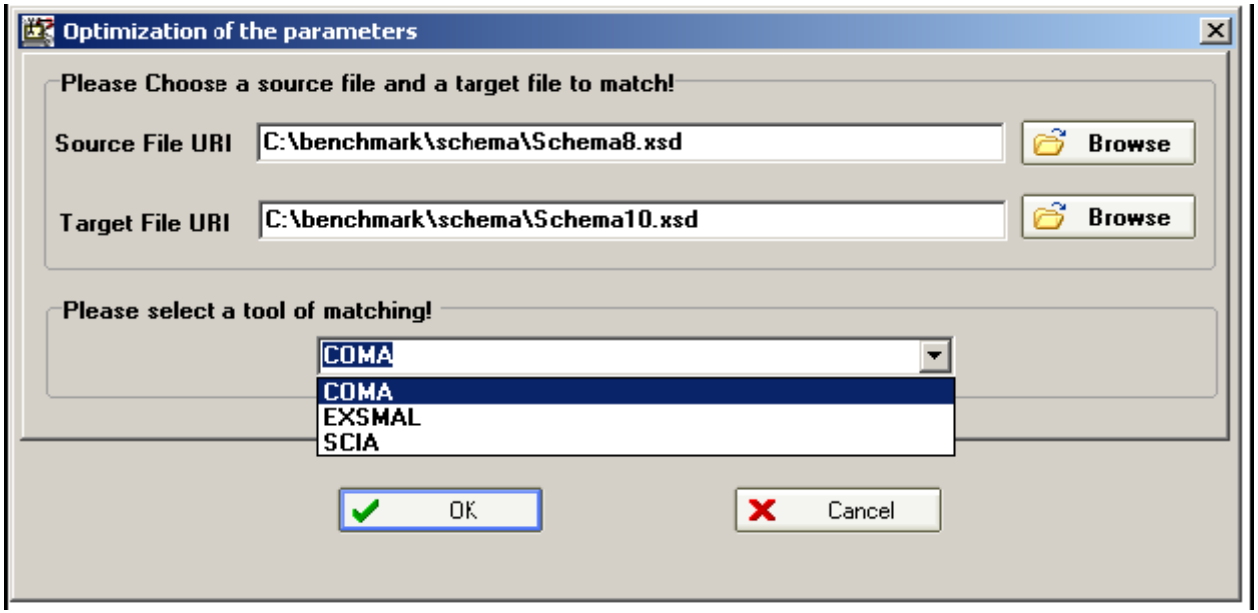
Notre approche



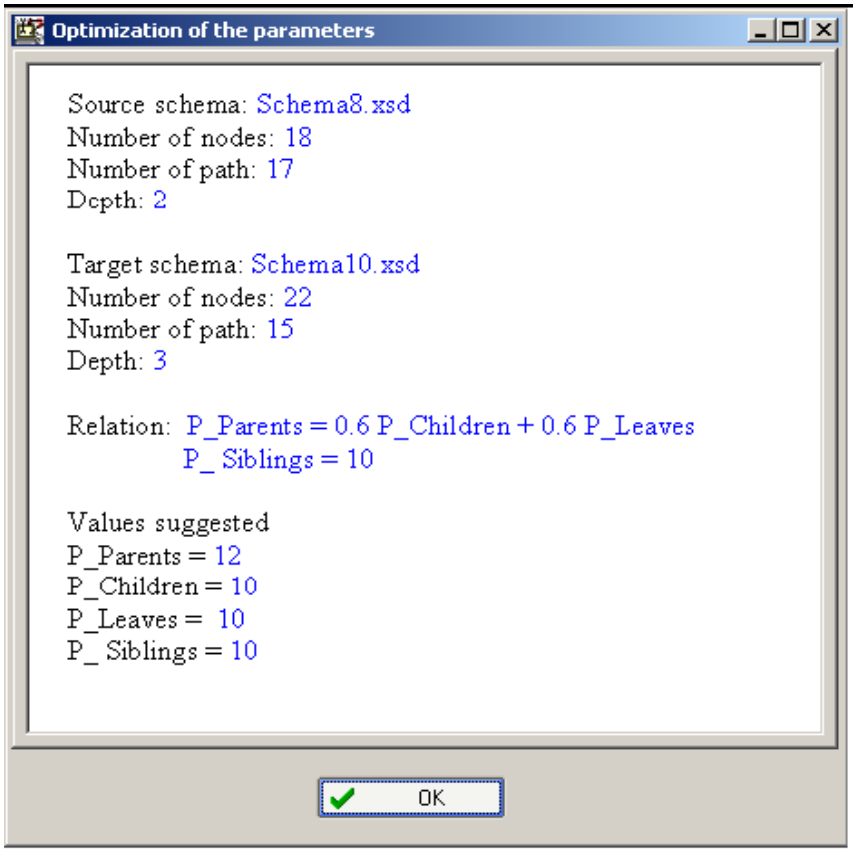
Algorithme de tuning



Algorithme de tuning

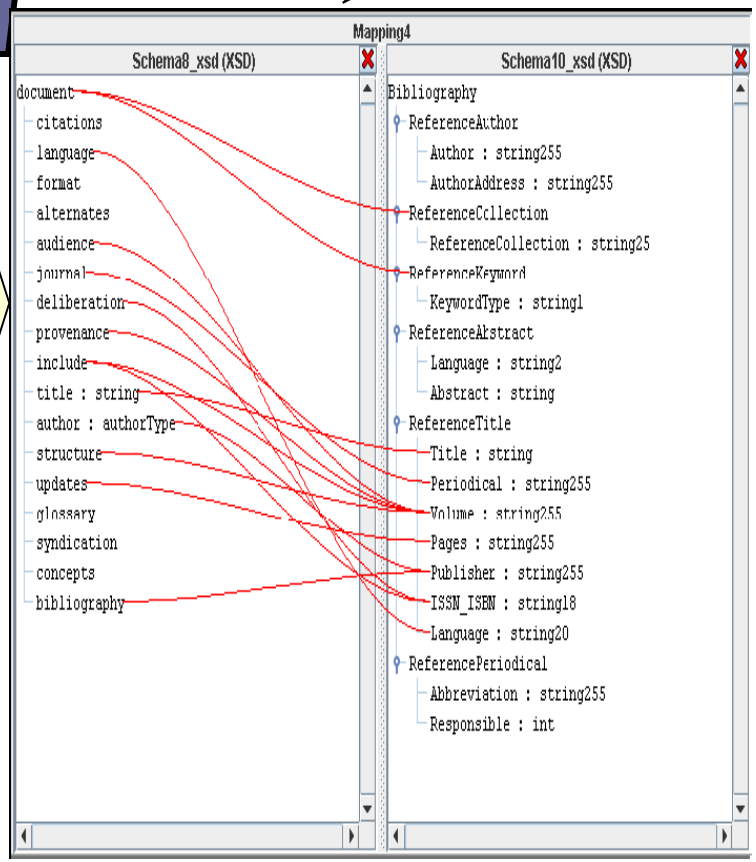


Algorithme de tuning



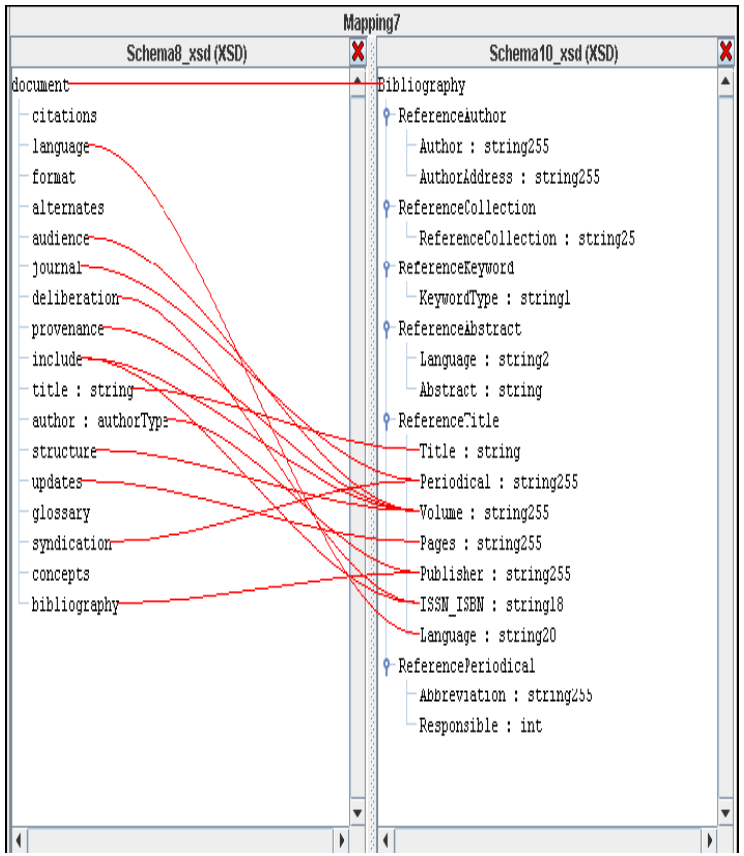
Algorithm tuning

Précision = 0,23



Le matching avant l'exécution de l'outil

Précision = 0,48



Le matching après l'exécution de l'outil

Conclusion

- Solution statistique
- Benchmark évolutif
- Algorithme de tuning évolutif
- Notre proposition par rapport à eTuner?

Perspectives

- Tuning des paramètres descriptifs
- Réaliser le benchmark avec un nombre de schémas plus important, et d'autre outils
- Intégrer cette approche dans la plateforme ASMADE

Je vous remercie de votre attention