Cubes Émergents pour l'analyse des renversements de tendances dans les bases de données multidimensionnelles

Sébastien NEDJAR

Laboratoire d'Informatique Fondamentale de Marseille

23 Novembre 2009

Plan

Contexte

- Les entrepôts de données permettent le stockage d'énormes volumes de données accumulées au fil du temps dans les bases opérationnelles.
- Les données sont dites multidimensionnelles car l'utilisateur voit et manipule ces données suivant différents critères, nommés *attributs dimensions*.
- Pendant le processus d'analyse, le décideur a besoin d'appréhender ses données suivant plusieurs combinaisons de ces dimensions.
- Pour que l'analyse soit interactive, il faut que la réponse à ces requêtes soit immédiate.

Contexte

- C'est pour répondre à cette problèmatique que le cube de données a été introduit.
- Le cube de données est constitué de l'ensemble de tous les agrégats possibles.
- Il permet d'extraire les grandes tendances présentes dans un jeu de données et d'en comprendre les origines.

Motivations

- Le cube de données et ses principales variantes isolent les tendances sur un seul jeu de données.
- Lors de la comparaison de deux cubes de données, l'information nouvelle provient des changements et des évolutions qui se sont produits.
- La connaissance des changements de tendances qui se produisent entre plusieurs jeux de données permet d'ajouter à l'analyse un aspect dynamique.
- Ce type de connaissances est particulièrement pertinent lorsqu'il s'agit d'étudier les données au cours du temps.

Relations exemples

Produit	Ville	Saison	Année	Quantité
1	Marseille	Printemps	2007	100
1	Marseille	Été	2007	100
2	Paris	Été	2007	100
3	Paris	Été	2007	100
2	Marseille	Printemps	2008	200
2	Paris	Été	2008	100
1	Marseille	Printemps	2008	100
3	Paris	Été	2008	100
3	Paris	Automne	2008	300

Relations exemples

Produit	Ville	Saison	Quantité
1	Marseille	Printemps	100
1	Marseille	Été	100
2	Paris	Été	100
3	Paris	Été	100

Produit	Ville	Saison	Quantité
2	Marseille	Printemps	200
2	Paris	Été	100
1	Marseille	Printemps	100
3	Paris	Été	100
3	Paris	Automne	300

Objectifs

- Isoler les changements de tendances les plus marqués : les renversements de tendances.
- Proposer une structure basée sur les concepts associés au treillis cube et aux motifs émergents pour extraire ce type de connaissances.
- Fournir un panel de représentations réduites qui soient adaptées aux besoins de l'utilisateur.
- Mettre au point des méthodes pour les rendre efficacement calculables.

Plan

Definition (Espace multidimensionnel)

L'espace multidimensionnel de la relation r regroupe toutes les combinaisons sémantiquement valides des ensembles de valeurs existant dans r pour les attributs dimensions, enrichis de la valeur ALL.

Definition (Ordre de généralisation)

Un tuple t est plus petit selon l'ordre de généralisation qu'un tuple u *ssi* t contient une information moins détaillée (plus agrégée) que u.

Exemple

(*ALL*, *Marseille*, *Printemps*)

Definition (Espace multidimensionnel)

L'espace multidimensionnel de la relation r regroupe toutes les combinaisons sémantiquement valides des ensembles de valeurs existant dans r pour les attributs dimensions, enrichis de la valeur ALL.

Definition (Ordre de généralisation)

Un tuple t est plus petit selon l'ordre de généralisation qu'un tuple u *ssi* t contient une information moins détaillée (plus agrégée) que u.

Exemple

$(ALL, Marseille, ALL) \preceq_g (ALL, Marseille, Printemps)$

La notion de renversement de tendances est formalisée par le concept d'émergence. Un tuple est dit « émergent » s'il est non significatif dans r_1 (contrainte C_1) mais très significatif dans r_2 (contrainte C_2).

Définition (Tuple Émergent)

un tuple $t \in CL(r_1 \cup r_2)$ est dit émergent de r_1 vers r_2 si et seulement s'il satisfait les deux contraintes C_1 et C_2 :

$$\begin{cases} f_{val}(t, r_1) < MinThreshold_1 \ (C_1) \\ f_{val}(t, r_2) \geq MinThreshold_2 \ (C_2) \end{cases}$$

Exemple

Soit $MinThreshold_1 = 200$ et $MinThreshold_2 = 200$. Le tuple $t = (ALL, Marseille, Printemps)$ est émergent.

La notion de renversement de tendances est formalisée par le concept d'émergence. Un tuple est dit « émergent » s'il est non significatif dans r_1 (contrainte C_1) mais très significatif dans r_2 (contrainte C_2).

Définition (Tuple Émergent)

un tuple $t \in CL(r_1 \cup r_2)$ est dit émergent de r_1 vers r_2 si et seulement s'il satisfait les deux contraintes C_1 et C_2 :

$$\begin{cases} f_{val}(t, r_1) < MinThreshold_1 \ (C_1) \\ f_{val}(t, r_2) \geq MinThreshold_2 \ (C_2) \end{cases}$$

Exemple

Soit $MinThreshold_1 = 200$ et $MinThreshold_2 = 200$. Le tuple $t = (ALL, Marseille, Printemps)$ est émergent.

Définition (Taux d'Emergence)

Pour caractériser la variation de tendance on utilise le taux d'émergence, noté $ER(t)$, défini par :

$$ER(t) = \begin{cases} 0 \text{ si } f_{val}(t, r_1) = 0 \text{ et } f_{val}(t, r_2) = 0 \\ \infty \text{ si } f_{val}(t, r_1) = 0 \text{ et } f_{val}(t, r_2) \neq 0 \\ \dfrac{f_{val}(t, r_2)}{f_{val}(t, r_1)} \text{ sinon.} \end{cases}$$

Tout tuple émergent a un taux d'émergence supérieur à $\frac{MinThreshold_2}{MinThreshold_1}$. Le choix des seuils permet à l'utilisateur de se focaliser uniquement sur les renversements de tendances qui l'intéressent.

Cube Émergent

Nous appelons Cube Émergent l'ensemble de tous les tuples de $CL(r_1 \cup r_2)$ émergents d'une relation r_1 vers autre relation r_2.

Définition (Cube Émergent)

Le Cube Émergent, noté $\mathbf{EC}(r_1,\ r_2)$, est défini par :

$$\mathbf{EC}(r_1,\ r_2) = \{t \in CL(r_1 \cup r_2) \mid C_1(t) \wedge C_2(t)\}$$

avec $C_1(t) = f_{val}(t, r_1) < MinThreshold_1$ et
$C_2(t) = f_{val}(t, r_2) \geq MinThreshold_2$.

Tuples Émergents	ER
(ALL, ALL , Printemps)	3
(ALL, Marseille , Printemps)	3
(2 , ALL , ALL)	3
(3 , ALL , ALL)	3
(3 , Paris , ALL)	3
(ALL, ALL , Automne)	∞
(ALL, Paris , Automne)	∞
(2 , ALL , Printemps)	∞
(2 , Marseille , ALL)	∞
(2 , Marseille , Printemps)	∞
(3 , ALL , Automne)	∞
(3 , Paris , Automne)	∞

Bordures Min/Max

Étant donnée la nature des contraintes d'émergence C_1 (monotone) et C_2 (anti-monotone), le Cube Émergent est un espace convexe. Il peut donc classiquement être représenté par deux bordures :

- La bordure des maximaux (les plus grands selon l'ordre de généralisation) satisfaisant la contrainte C_2 notée U.
- La bordure des minimaux (les plus petits selon l'ordre de généralisation) satisfaisant la contrainte C_1 et C_2 notée L.

Définition (Bordures *[L ;U]*)

$$\begin{cases} L = \min_{\preceq_g}(\{t \in CL(r_1 \cup r_2) \mid C_1(t) \wedge C_2(t)\}) \\ U = \max_{\preceq_g}(\{t \in CL(r_1 \cup r_2) \mid C_1(t) \wedge C_2(t)\}) \end{cases}$$

Bordures Min/Max

Étant donnée la nature des contraintes d'émergence C_1 (monotone) et C_2 (anti-monotone), le Cube Émergent est un espace convexe. Il peut donc classiquement être représenté par deux bordures :

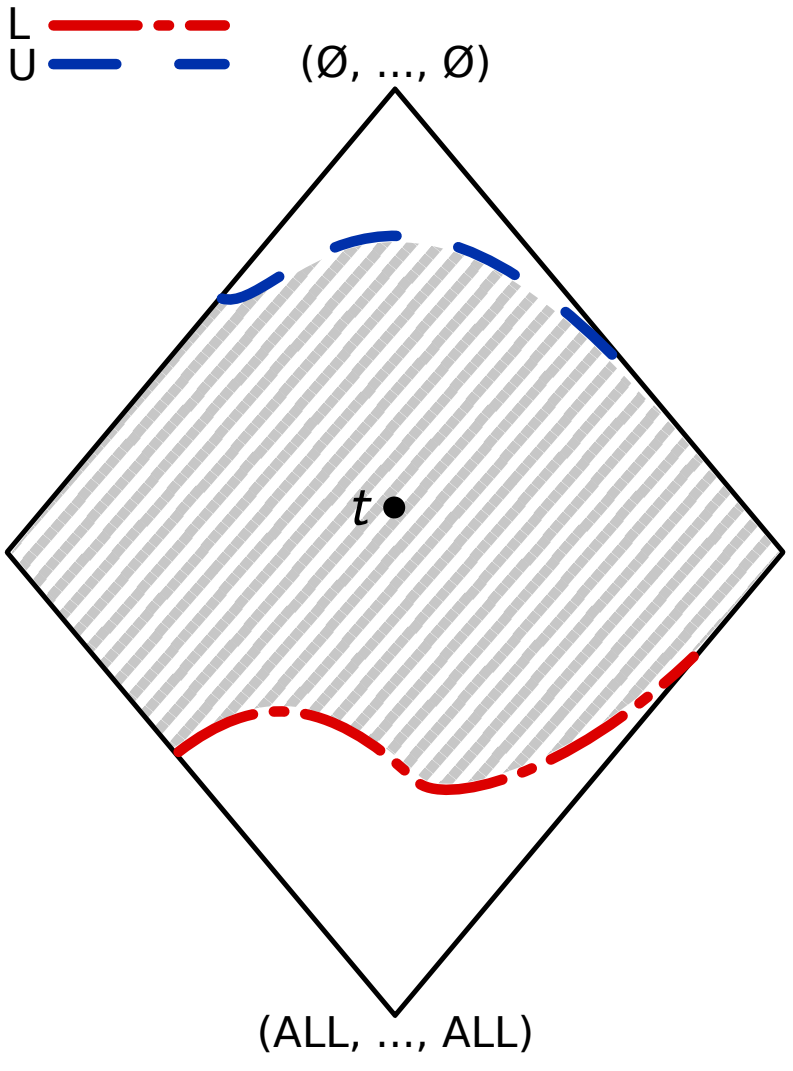
- La bordure des maximaux (les plus grands selon l'ordre de généralisation) satisfaisant la contrainte C_2 notée U.
- La bordure des minimaux (les plus petits selon l'ordre de généralisation) satisfaisant la contrainte C_1 et C_2 notée L.

Définition (Bordures *[L ;U]*)

$$\begin{cases} L = \min_{\preceq_g}(\{t \in CL(r_1 \cup r_2) \mid C_1(t) \wedge C_2(t)\}) \\ U = \max_{\preceq_g}(\{t \in CL(r_1 \cup r_2) \mid C_1(t) \wedge C_2(t)\}) \end{cases}$$

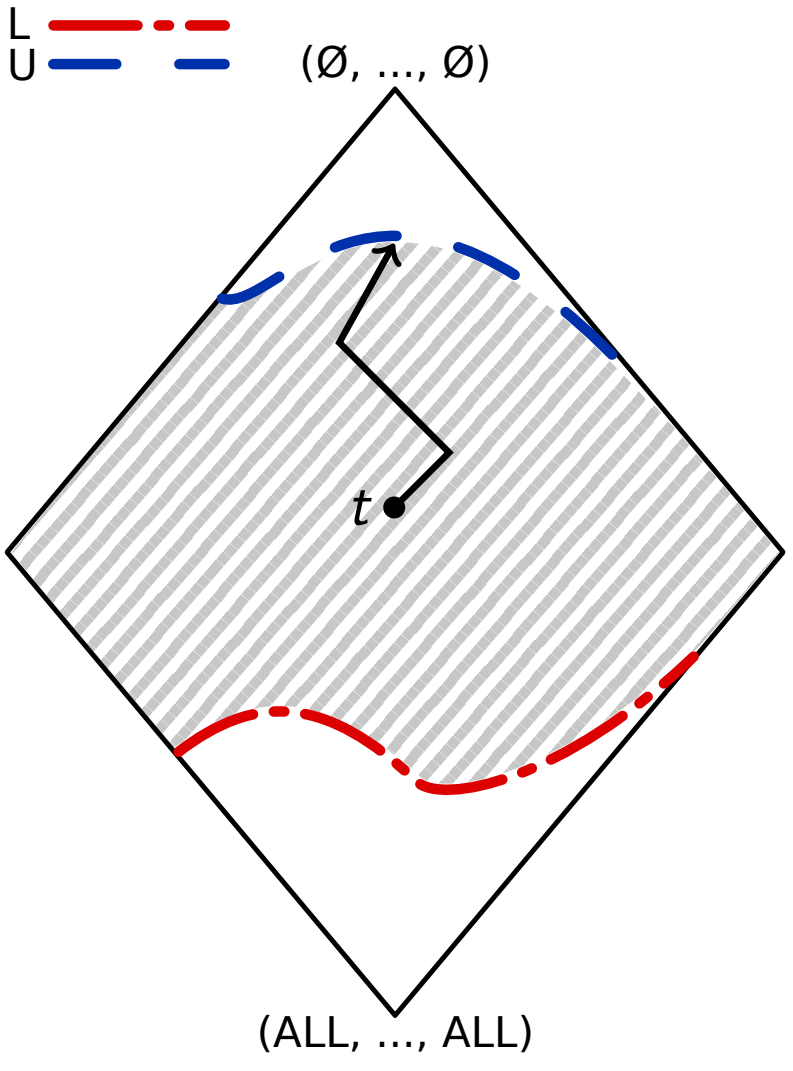
Un tuple t est émergent si et seulement si :

1. il généralise un tuple de U (satisfait C_2)
2. il spécialise un tuple de L (satisfait C_1 et C_2)



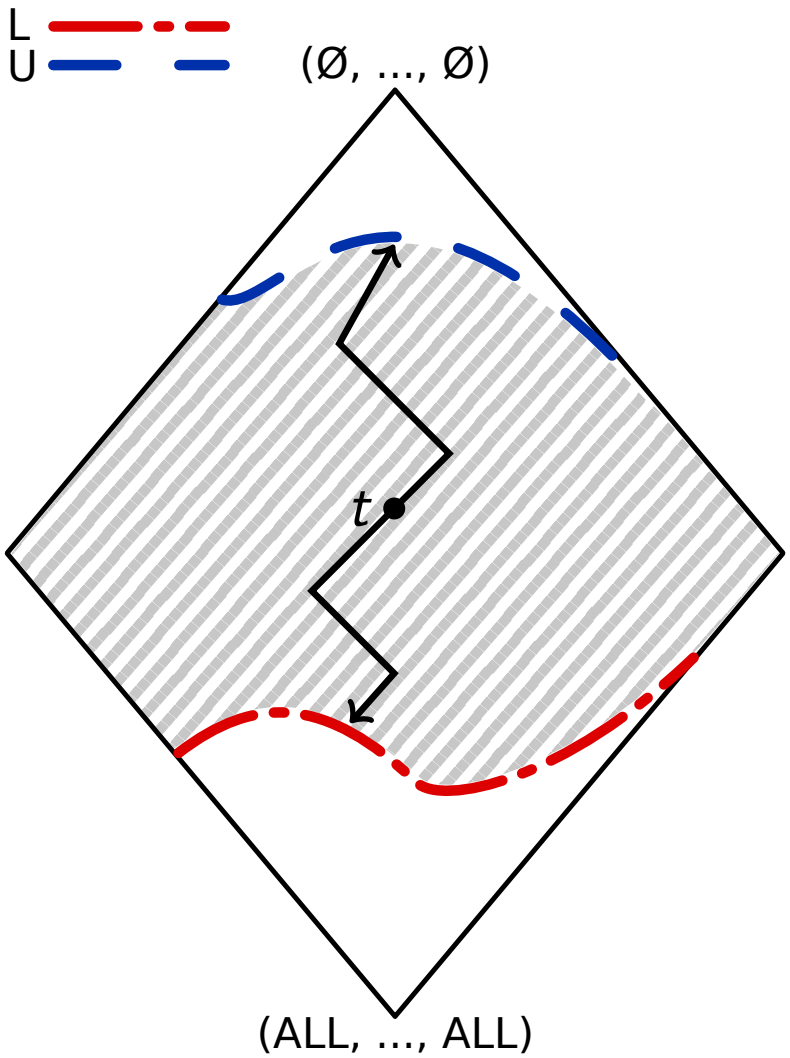
Un tuple t est émergent si et seulement si :

1. il généralise un tuple de U (satisfait C_2)
2. il spécialise un tuple de L (satisfait C_1 et C_2)



Un tuple t est émergent si et seulement si :

1. il généralise un tuple de U (satisfait C_2)
2. il spécialise un tuple de L (satisfait C_1 et C_2)



Bordure U du Cube Émergent de nos relations exemples :

(2 , Marseille , Printemps)
(3 , Paris , Automne)

Bordure L du Cube Émergent de nos relations exemples :

(ALL, ALL , Printemps)
(2 , ALL , ALL)
(3 , ALL , ALL)
(ALL, ALL , Automne)

Exemple

Bordure U du Cube Émergent de nos relations exemples :

(2 , Marseille , Printemps)
(3 , Paris , Automne)

Bordure L du Cube Émergent de nos relations exemples :

(ALL, ALL , Printemps)
(2 , ALL , ALL)
(3 , ALL , ALL)
(ALL, ALL , Automne)

Exemple

Le tuple (ALL, Marseille, Printemps) est émergent car il généralise (2, Marseille, Printemps) et il spécialise (ALL, ALL, Printemps)

Bordure U du Cube Émergent de nos relations exemples :

(2 , Marseille , Printemps)
(3 , Paris , Automne)

Bordure L du Cube Émergent de nos relations exemples :

(ALL, ALL , Printemps)
(2 , ALL , ALL)
(3 , ALL , ALL)
(ALL, ALL , Automne)

Exemple

Le tuple (ALL, Marseille, ALL) n'est pas émergent car il ne spécialise aucun tuple de L

Bordures Max/Max

La négation d'une contrainte monotone est une contrainte anti-monotone. La contrainte $\neg C_1$ est représentable par l'ensemble des plus grands tuples (selon l'ordre de généralisation) ne satisfaisant pas C_1.

Grâce à cette propriété, nous proposons une nouvelle bordure ($U^\sharp$) qui se substitue à L.

Définition (Bordures $]U^\sharp ; U]$)

$$\begin{cases} U^\sharp = \max_{\preceq_g}(\{t \in CL(r_2) \mid \neg C_1(t) \wedge C_2(t)\}) \\ U = \max_{\preceq_g}(\{t \in CL(r_2) \mid C_1(t) \wedge C_2(t)\}) \end{cases}$$

Bordures Max/Max

La négation d'une contrainte monotone est une contrainte anti-monotone. La contrainte $\neg C_1$ est représentable par l'ensemble des plus grands tuples (selon l'ordre de généralisation) ne satisfaisant pas C_1.

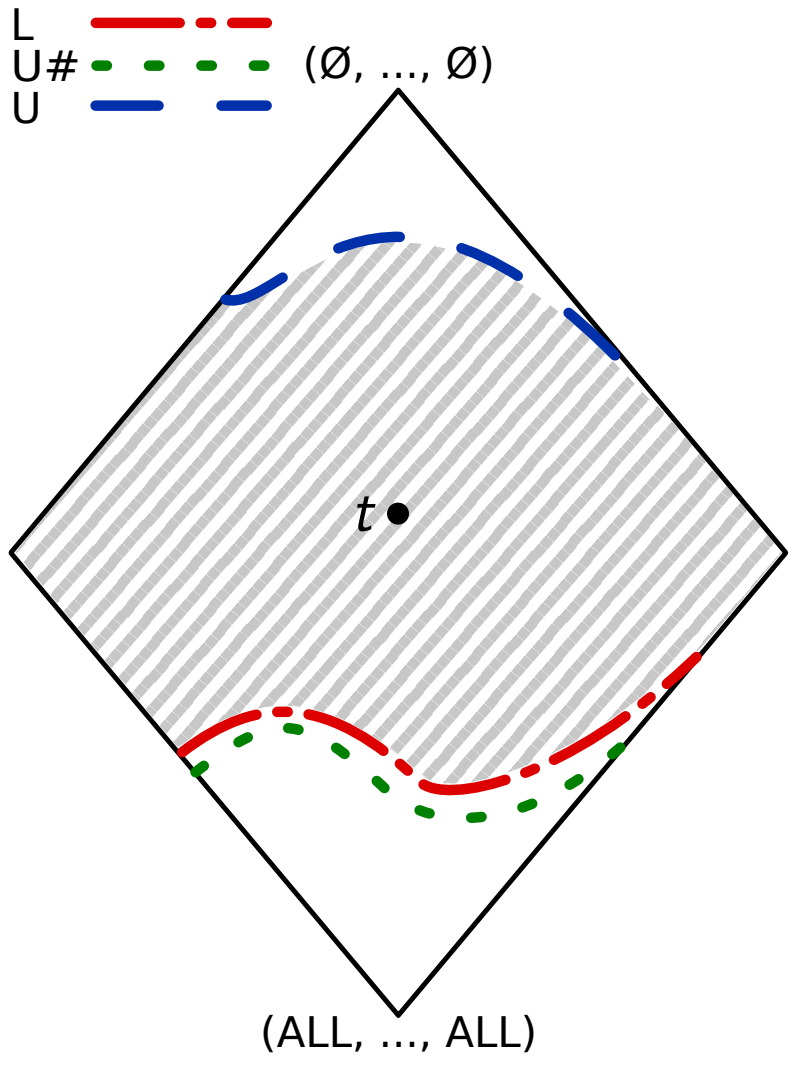
Grâce à cette propriété, nous proposons une nouvelle bordure ($U^\sharp$) qui se substitue à L.

Définition (Bordures $]U^\sharp ; U]$)

$$\begin{cases} U^\sharp = \max_{\preceq_g}(\{t \in CL(r_2) \mid \neg C_1(t) \wedge C_2(t)\}) \\ U = \max_{\preceq_g}(\{t \in CL(r_2) \mid C_1(t) \wedge C_2(t)\}) \end{cases}$$

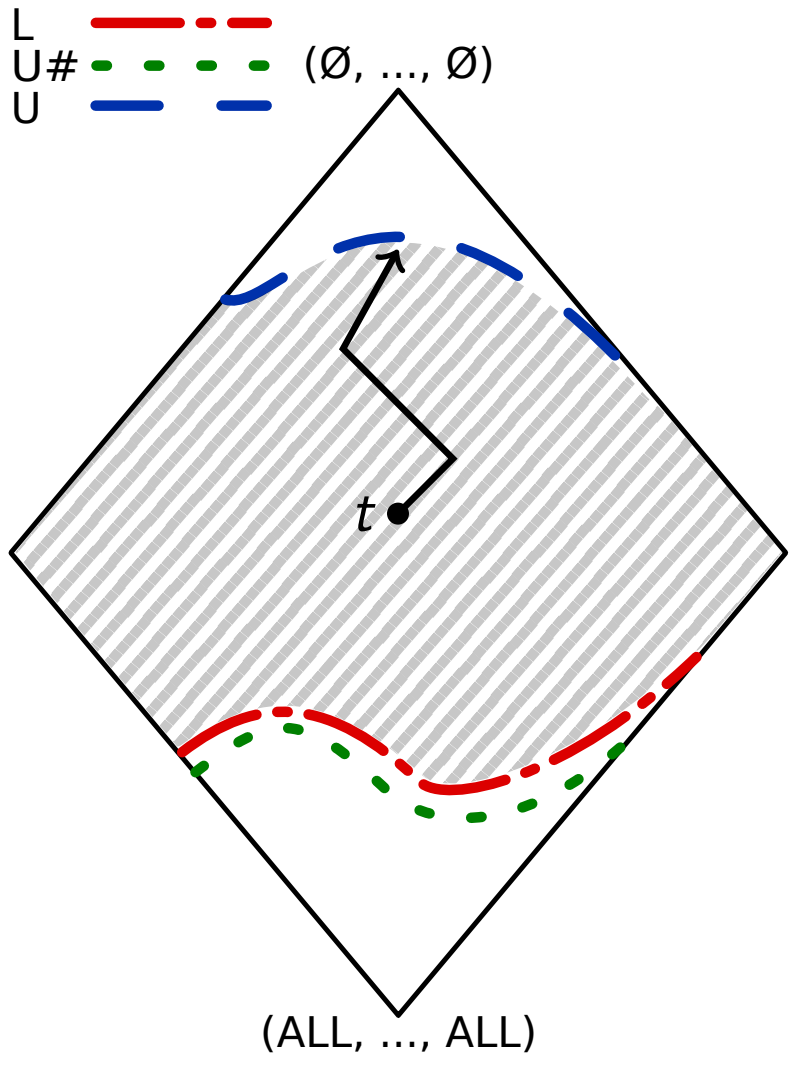
Un tuple t est émergent si et seulement si :

1. il généralise un tuple de U (satisfait C_2)
2. il ne généralise aucun tuple de $U^\sharp$ (ne satisfait pas $\neg C_1$ et satisfait C_2)



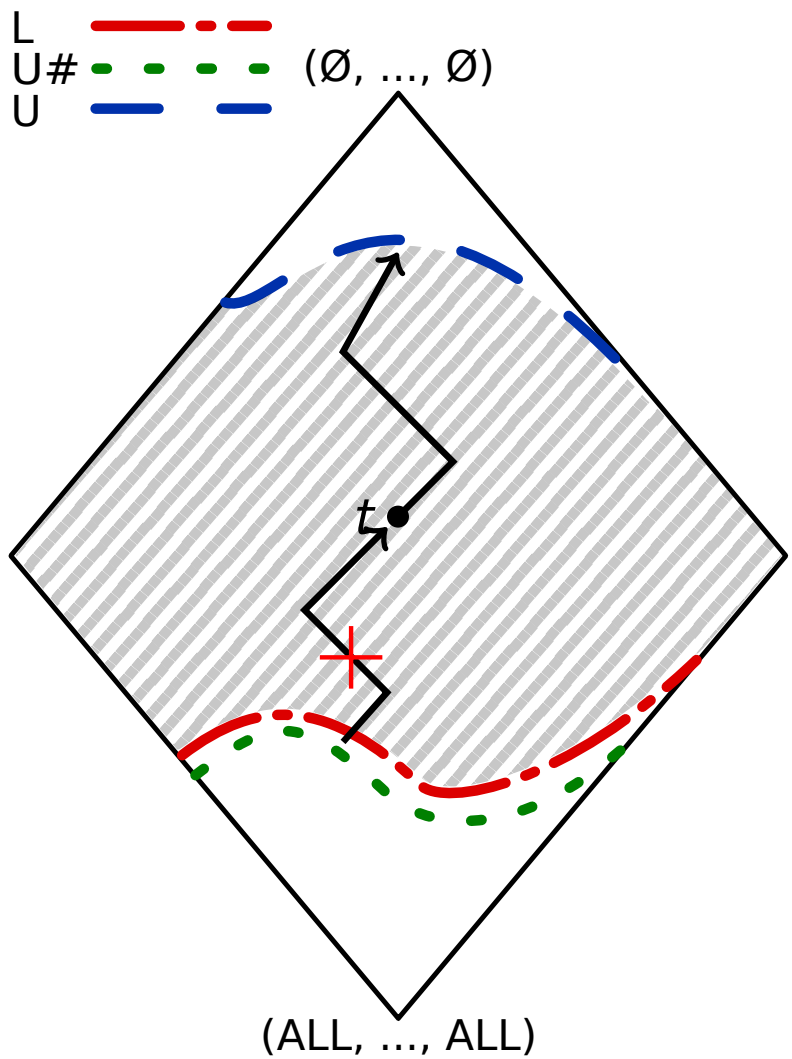
Un tuple t est émergent si et seulement si :

1. il généralise un tuple de U (satisfait C_2)
2. il ne généralise aucun tuple de $U^\sharp$ (ne satisfait pas $\neg C_1$ et satisfait C_2)



Un tuple t est émergent si et seulement si :

1. il généralise un tuple de U (satisfait C_2)
2. il ne généralise aucun tuple de $U^\sharp$ (ne satisfait pas $\neg C_1$ et satisfait C_2)



Bordure U du Cube Émergent de nos relations exemples :

(2 , Marseille , Printemps)
(3 , Paris , Automne)

Bordure $U^{\sharp}$ du Cube Émergent de nos relations exemples :

(ALL , Marseille , ALL)
(ALL , Paris , ALL)

Exemple

Bordure U du Cube Émergent de nos relations exemples :

(2 , Marseille , Printemps)
(3 , Paris , Automne)

Bordure $U^{\sharp}$ du Cube Émergent de nos relations exemples :

(ALL , Marseille , ALL)
(ALL , Paris , ALL)

Exemple

Le tuple (ALL, Marseille, Printemps) est émergent car il généralise (2, Marseille, Printemps) et ne généralise ni (ALL , Marseille , ALL) ni (ALL , Paris , ALL)

Bordure U du Cube Émergent de nos relations exemples :

(2 , Marseille , Printemps)
(3 , Paris , Automne)

Bordure $U^{\sharp}$ du Cube Émergent de nos relations exemples :

(ALL , Marseille , ALL)
(ALL , Paris , ALL)

Exemple

Le tuple (ALL, Marseille, ALL) n'est pas émergent car il généralise un tuple de $U^{\sharp}$

Intérêts des bordures Max/Max

- Utilisent deux bordures de même nature (des maximaux) donc un seul et même algorithme pour calculer le couple de bordures.
- Expérimentalement bien plus réduites que les bordures Min/Max.
- Nous ont permis de proposer une caractérisation naturelle de la taille du Cube Émergent.
- Grâce à ces bordures nous avons proposé une représentation sans perte d'information extrêmement réduite.

Constats

- Points négatifs :
  - Le problème de comptages des tuples émergent est $\sharp P$-Complet.
  - L'énumération des tuples émergents et l'extraction des bordures du Cube Émergent sont NP-Difficiles.
- Point positif :
  - Dans le contexte **OLAP** le nombre de dimensions est raisonnable ($3 \leq |Dim| \leq 20$).
- Cela nous enseigne :
  - Dans le cas général, les algorithmes de calcul du Cube Émergent (ou de ses bordures) sont inabordables. Mais dans notre cas on peut considérer le problème comme traitable.

Constats

- Points négatifs :
  - Le problème de comptages des tuples émergent est $\sharp P$-Complet.
  - L'énumération des tuples émergents et l'extraction des bordures du Cube Émergent sont NP-Difficiles.
- Point positif :
  - Dans le contexte **OLAP** le nombre de dimensions est raisonnable ($3 \leq |Dim| \leq 20$).
- Cela nous enseigne :
  - Dans le cas général, les algorithmes de calcul du Cube Émergent (ou de ses bordures) sont inabordables. Mais dans notre cas on peut considérer le problème comme traitable.

Constats

- Points négatifs :
  - Le problème de comptages des tuples émergent est $\sharp P$-Complet.
  - L'énumération des tuples émergents et l'extraction des bordures du Cube Émergent sont NP-Difficiles.
- Point positif :
  - Dans le contexte **OLAP** le nombre de dimensions est raisonnable ($3 \leq |Dim| \leq 20$).
- Cela nous enseigne :
  - Dans le cas général, les algorithmes de calcul du Cube Émergent (ou de ses bordures) sont inabordables. Mais dans notre cas on peut considérer le problème comme traitable.

Plan

Taille du Cube Émergent

Plus le Cube Émergent est petit par rapport au cube de r_2 :

- Plus l'information capturée a une forte sémantique (renversement de tendances les plus pertinents).
- Plus il sera facilement appréhendable et manipulable par l'utilisateur.
- Plus une approche algorithmique dédiée sera performante.

Connaître la taille *a priori* permet :

- Pour l'utilisateur de calibrer les contraintes pour obtenir un résultat approprié sans faire de calculs inutiles et couteux.
- Pour le système de choisir l'approche algorithmique la plus adaptée.
- Pour l'administrateur prévoir l'espace de stockage nécessaire pour le Cube Émergent.

Problème du calcul

Le problème est $\sharp P$-Complet : compter le nombre de tuples émergents est difficile.

- La seule solution est de parcourir (énumérer) l'espace de recherche et pour chaque tuple tester son émergence.
- Plus le nombre de tests est minimisé plus le comptage sera efficace

Les bordures résument l'ensemble des tuples émergents. On peut s'en servir pour réduire le nombre de tests.

Caractérisation de la taille exacte

Définition (Idéal d'un Ordre)

Soit $T \subseteq CL(r)$ un ensemble de tuples. Un idéal d'ordre généré par T est noté $\downarrow T$ et comprend tous les tuples généralisant au moins un tuple de T. $\downarrow T$ est défini comme suit :

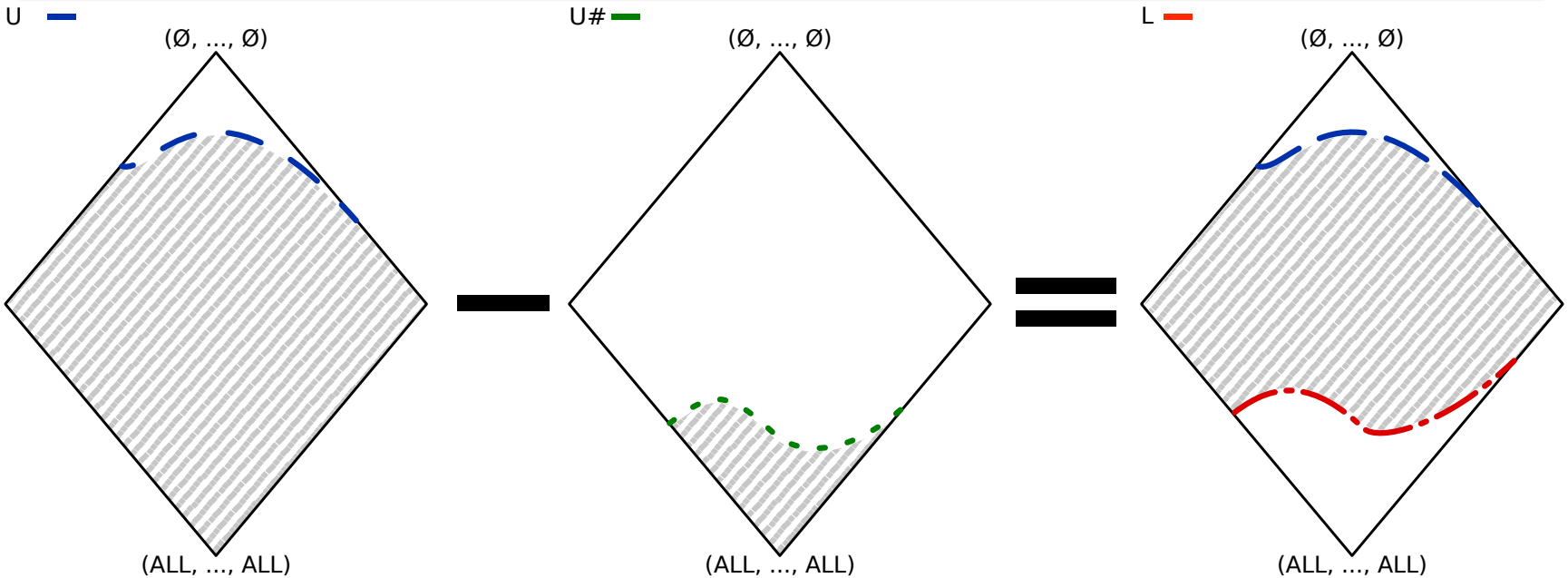
$$\downarrow T = \{t \in CL(r) \mid \exists t' \in T \mid t \preceq_g t'\}$$

L'ensemble des tuples satisfaisant une contrainte anti-monotone forme un idéal d'ordre. Cet idéal peut être généré à partir des tuples maximaux satisfaisant la contrainte.

Proposition (Taille du Cube Émergent)

Soit $]U^{\sharp}, U]$ les bordures du Cube Émergent $\mathbf{EC}(r_1, r_2)$. La taille de ce dernier peut être exprimée de la manière suivante :

$$|\mathbf{EC}(r_1, r_2)| = |\downarrow U| - |\downarrow U^{\sharp}|$$



Caractérisation de la taille exacte

- Enumérer les éléments d'un idéal d'ordre peut se faire directement. Avec cette caractérisation on peut calculer la taille exacte d'un Cube Émergent sans faire aucun test d'émergence.
- Compter le nombre exact d'éléments distincts (la cardinalité) d'un multi-ensemble demande de conserver en mémoire la liste de tous les éléments déjà rencontrés.
- Dans notre cas, cette liste ne tient pas en mémoire centrale. Le grand nombre d'accès disque générés par les accès aléatoires à cette liste a un coût prohibitif (temps).

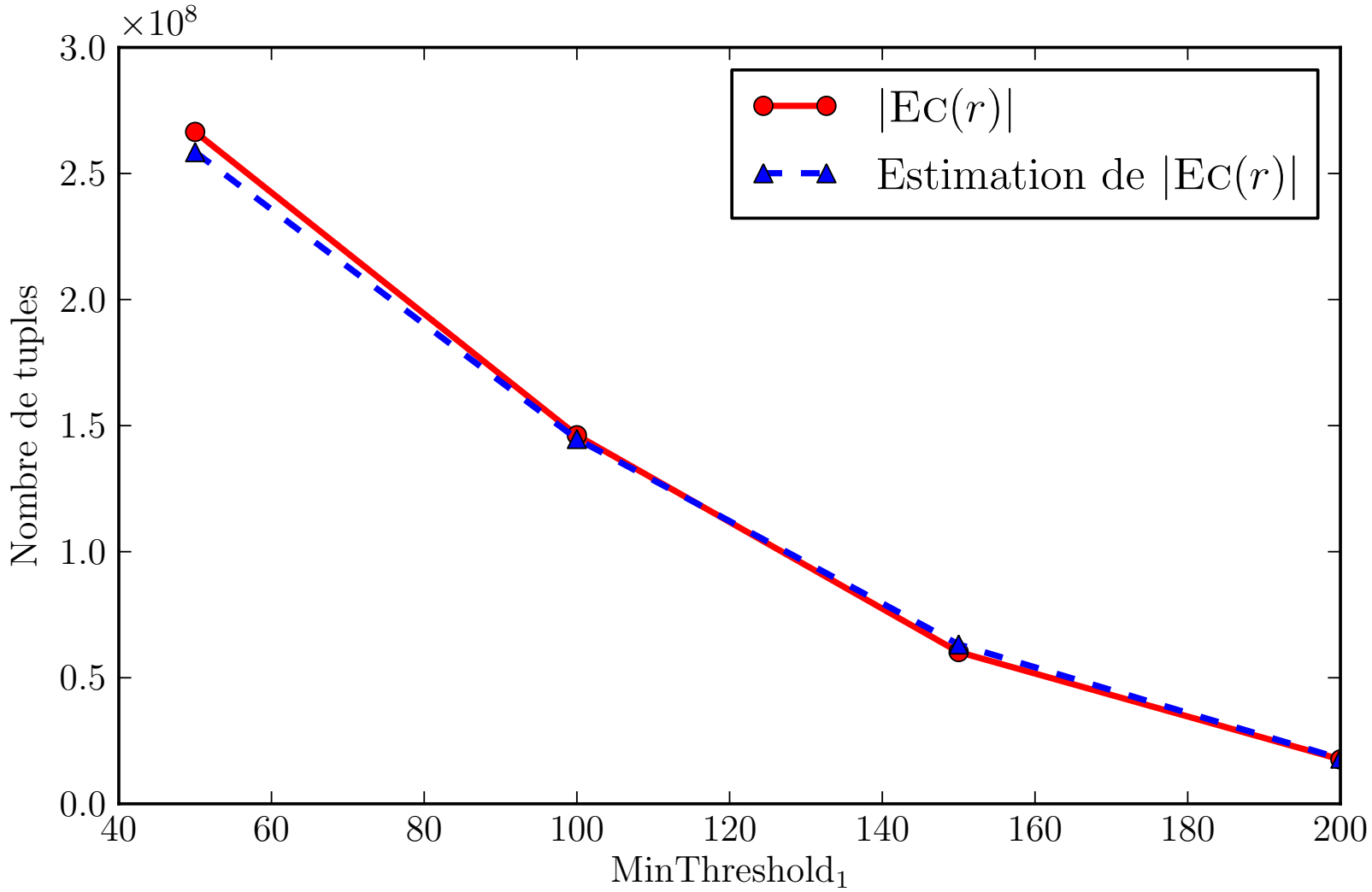
Estimation

- Calculer la taille exacte du Cube Émergent consomme trop de mémoire. Les nombreux accès disque générés rendent ce calcul impratiquable.
- Connaître la taille exacte n'est pas indispensable. La contrainte d'exactitude peut être relachée, une bonne approximation suffit.
- Nous proposons un nouvel algorithme **Idea-LogLog** qui se base sur le meilleur algorithme d'estimation de la cardinalité d'un multi-ensemble connu HyperLogLog.

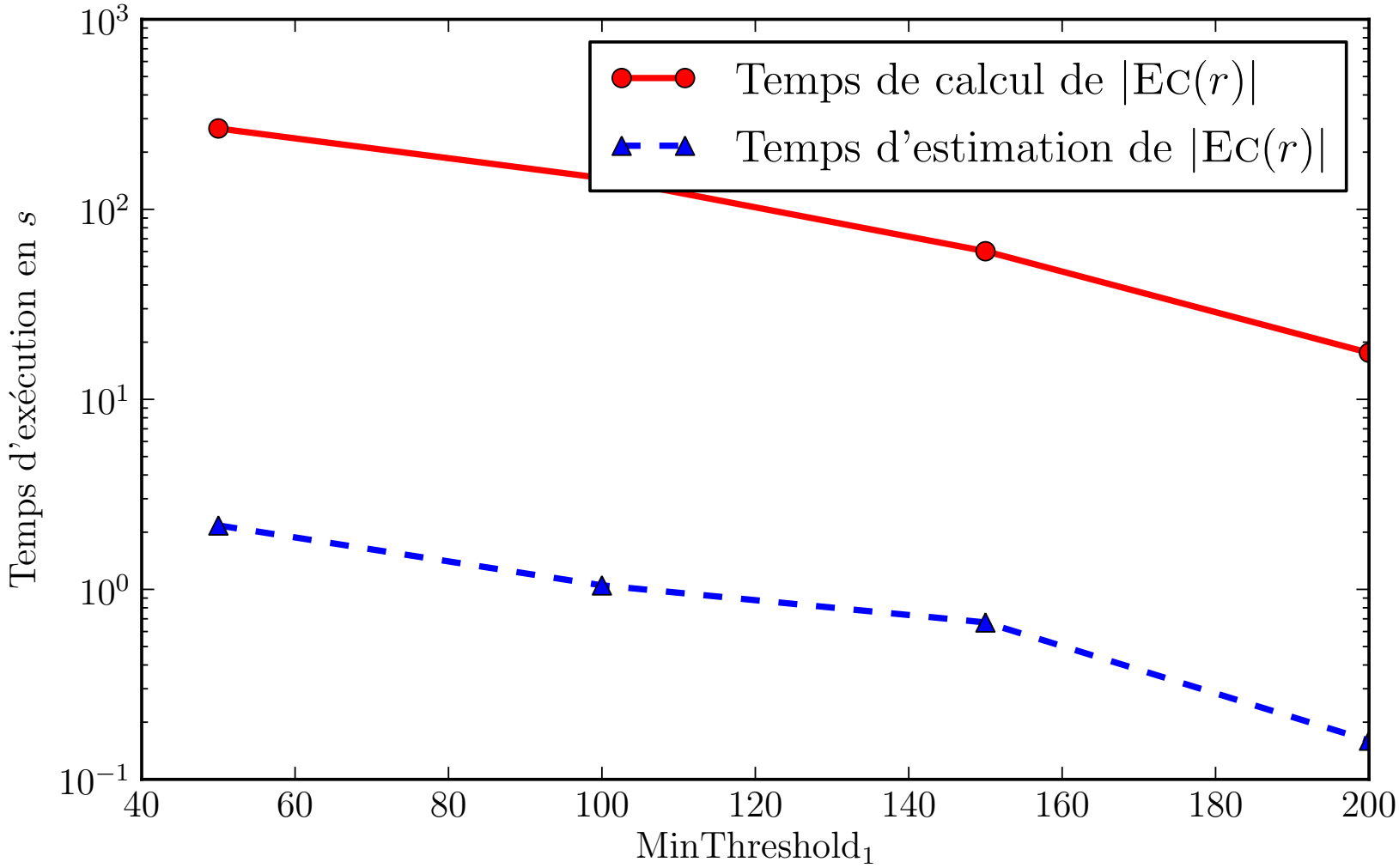
Intérêts

- Cet algorithme ne demande qu'un seul passage sur les données. Donc les coûts d'accès disque sont réduits au minimum.
- Le temps de réponse dépend principalement du nombre de dimensions. Ce nombre étant dans notre contexte borné ($3 \leq |Dim| \leq 20$), l'estimation de la taille est efficace.
- Complexité mémoire très faible (en $O(\log \log N)$) avec 1024 octets on peut estimer des cardinalités $> 10^9$
- Expérimentalement, on a constaté que l'estimation est tout à fait correcte (taux d'erreur inférieur à 5%) et dans un temps plusieurs ordres de grandeur inférieur au temps de calcul du Cube Émergent complet.

Tailles exacte et approximative du Cube Émergent pour les données météorologiques :



Estimation et temps de calcul du Cube Émergent pour les données météorologiques :



Plan

Représentations du Cube Émergent

- Dans un contexte **OLAP**, la valeur du taux d'émergence doit être retrouvée. Les bordures ne suffisent plus.
- Le Cube Émergent a une taille du même ordre de grandeur que le data cube complet. Il est donc important d'avoir une structure la plus réduite possible, pour restreindre le problème du stockage.
- En se basant sur le travail fait sur les bordures nous proposons plusieurs représentations sans perte d'information (taux d'émergence).

Cubes Fermés Émergents

Définition (Fermeture Cubique)

Soit $T \subseteq CL(r)$ un ensemble de tuples, l'opérateur de Fermeture Cubique $\mathbb{C} : CL(r) \rightarrow CL(r)$ selon T peut être défini comme suit :

$$\mathbb{C}(t, T) = (\emptyset, \ldots, \emptyset) + \sum_{t' \in T, t \preceq_g t'} t'$$

Un tuple t est dit fermé sur r si et seulement si $\mathbb{C}(t, r) = t$.

L'ensemble des tuples fermés d'une relation r forme une couverture du cube de données pour les fonctions mesures compatibles avec la fermeture (comme *SUM* ou *COUNT*) grâce à la propriété suivante :

$$f_{val}(t, r) = f_{val}(\mathbb{C}(t, r), r)$$

Définition (Couverture du Cube Émergent)

Un ensemble de tuples T est une couverture du Cube Émergent (basée sur la fermeture cubique) *ssi* $\forall t \in CL(r_1 \cup r_2)$:

1. on peut décider si t est un tuple émergent.
2. si t est émergent alors $\mathbb{C}(t, T) = \mathbb{C}(t, r_1 \cup r_2)$

Définition (Tuple Fermé Émergent)

Soit $t \in CL(r_1 \cup r_2)$ un tuple, t est un tuple fermé émergent *ssi* :

1. t est un tuple émergent ;
2. $\mathbb{C}(t, r_1 \cup r_2) = t$.

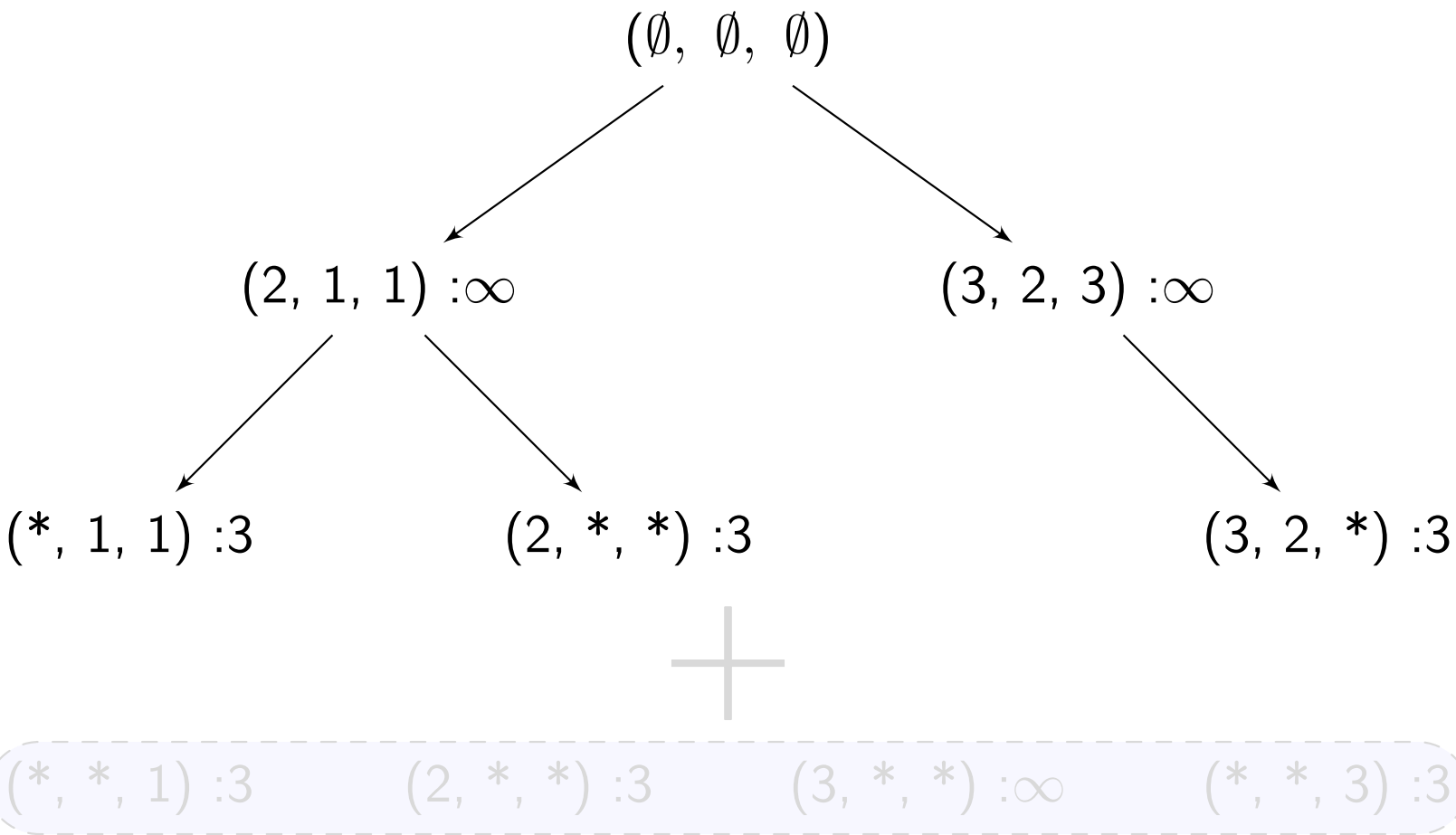
L-Cubes Fermés Émergents

- L'ensemble des tuples fermés émergents n'est pas une couverture du Cube Émergent car même si on peut calculer correctement la fermeture, il existe des tuples pour lesquels on ne peut pas décider s'ils sont émergents ou non.
- Afin de résoudre le problème d'appartenance et ainsi obtenir une couverture du Cube Émergent (basée sur la fermeture cubique), nous combinons les tuples fermés émergents avec les bordures.

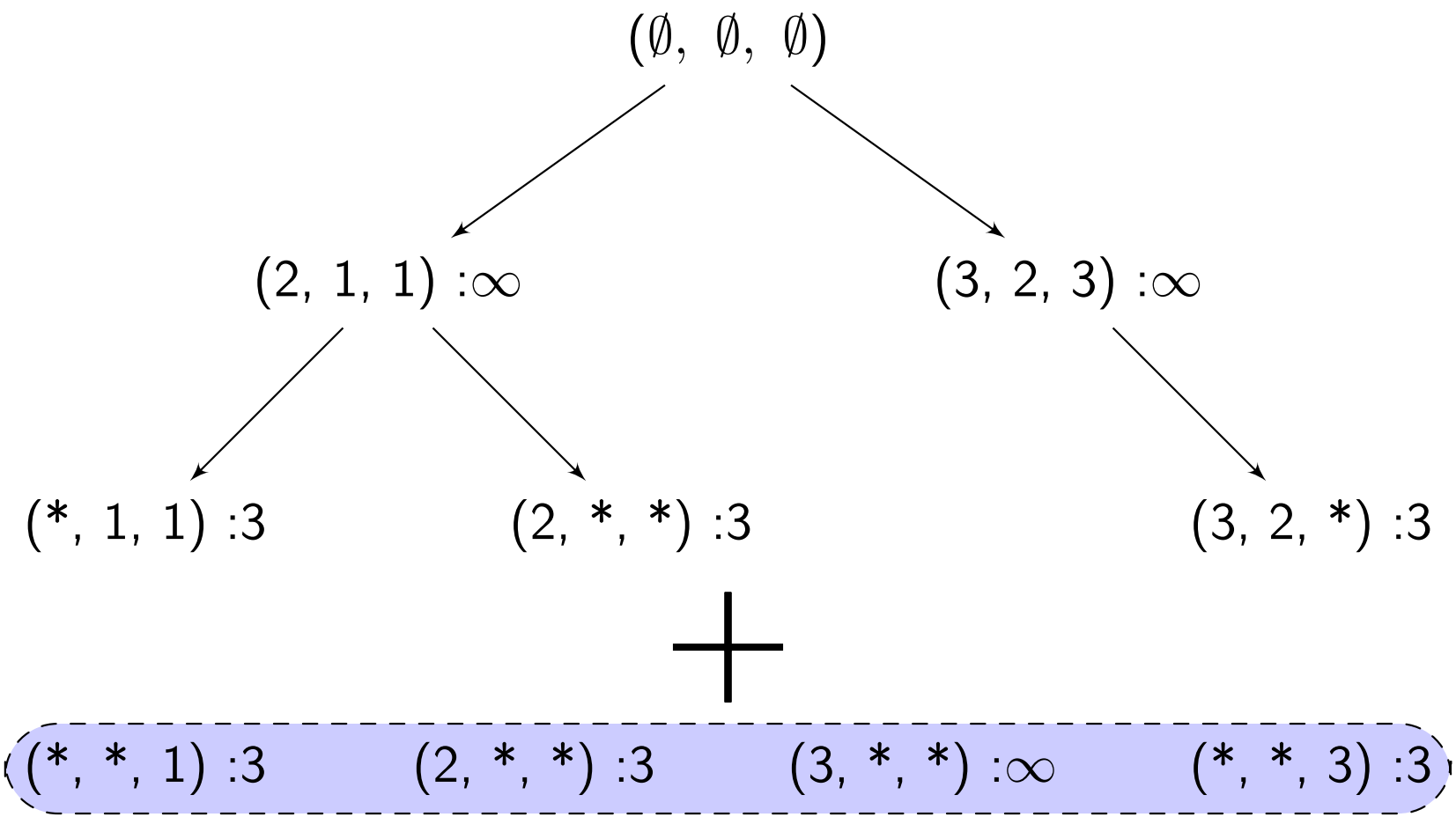
Définition (L-Cube Fermé Émergent)

L-ECC$(r_1, r_2) = \{t \in CL(r_1 \cup r_2) \mid$
t est un tuple fermé émergent$\} \cup L$

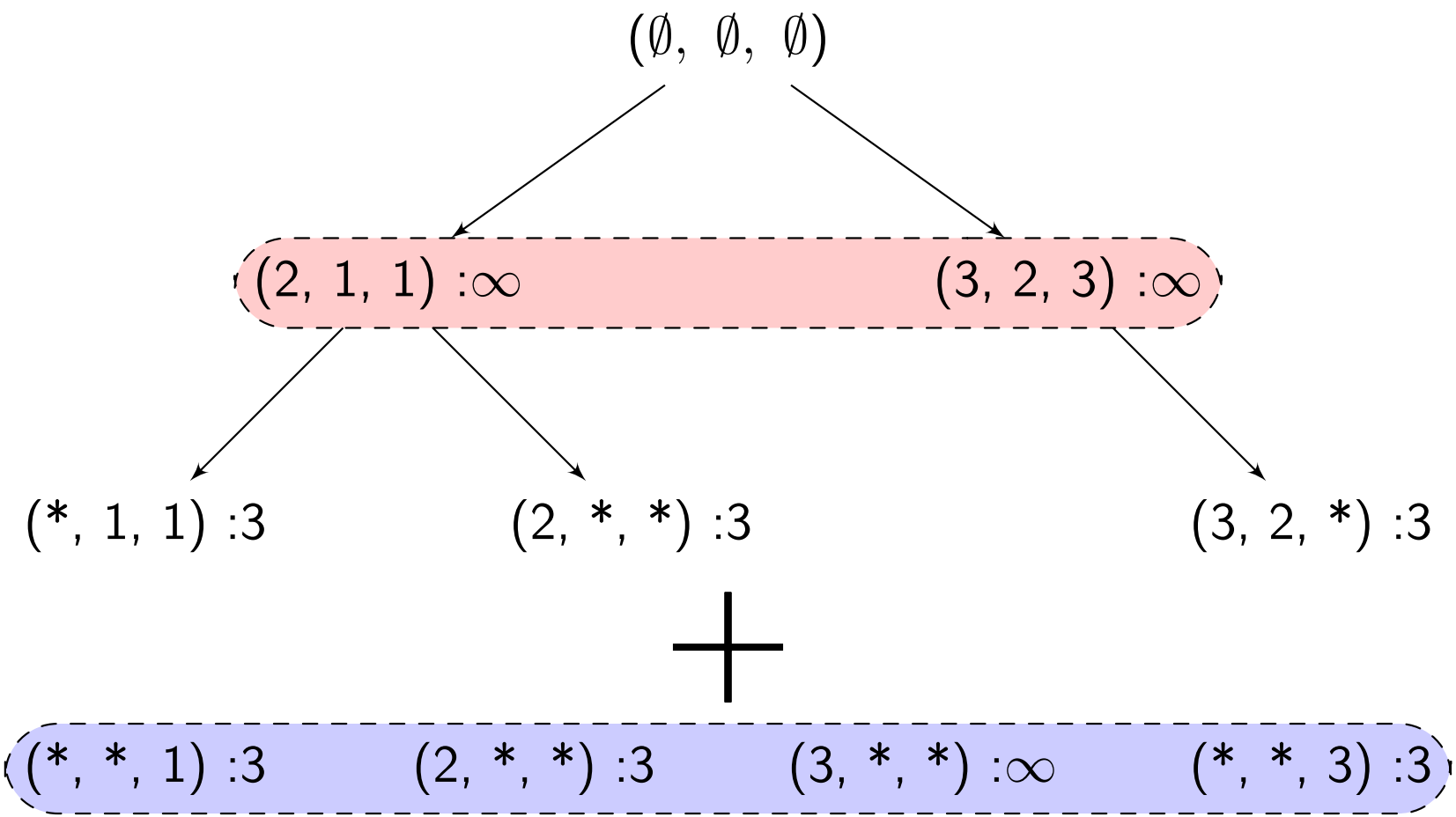
L-Cube Fermé Émergent de notre exemple :



L-Cube Fermé Émergent de notre exemple :



L-Cube Fermé Émergent de notre exemple :



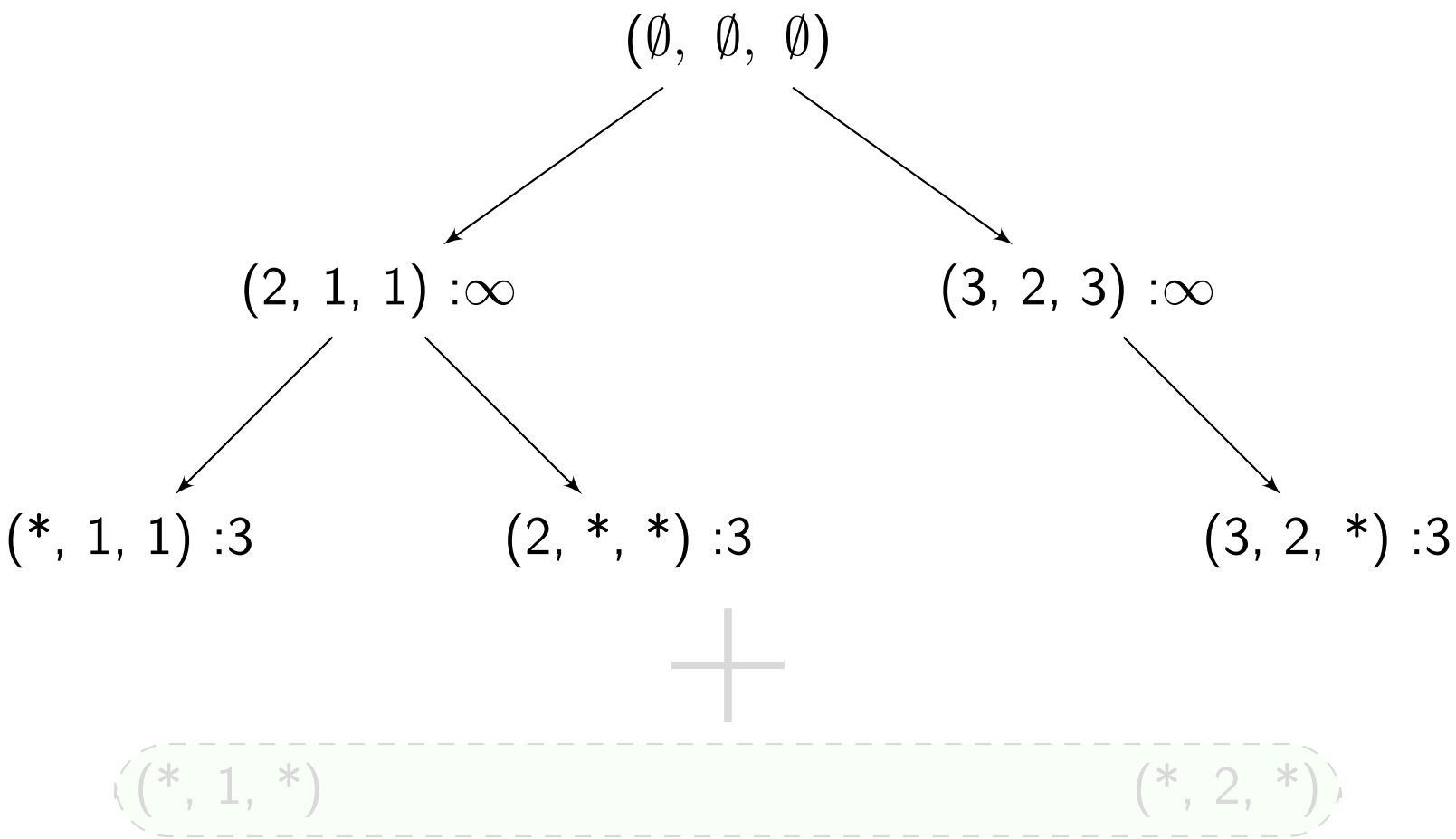
$U^\sharp$-Cubes Fermés Émergents

- La bordure L ne contient pas que des tuples fermés. Pour obtenir une couverture uniquement à base de fermés on remplace la bordure L par $U^\sharp$.
- La nouvelle représentation bénéficie de tous les avantages de la bordure $U^\sharp$ (compacité et calcul efficace)

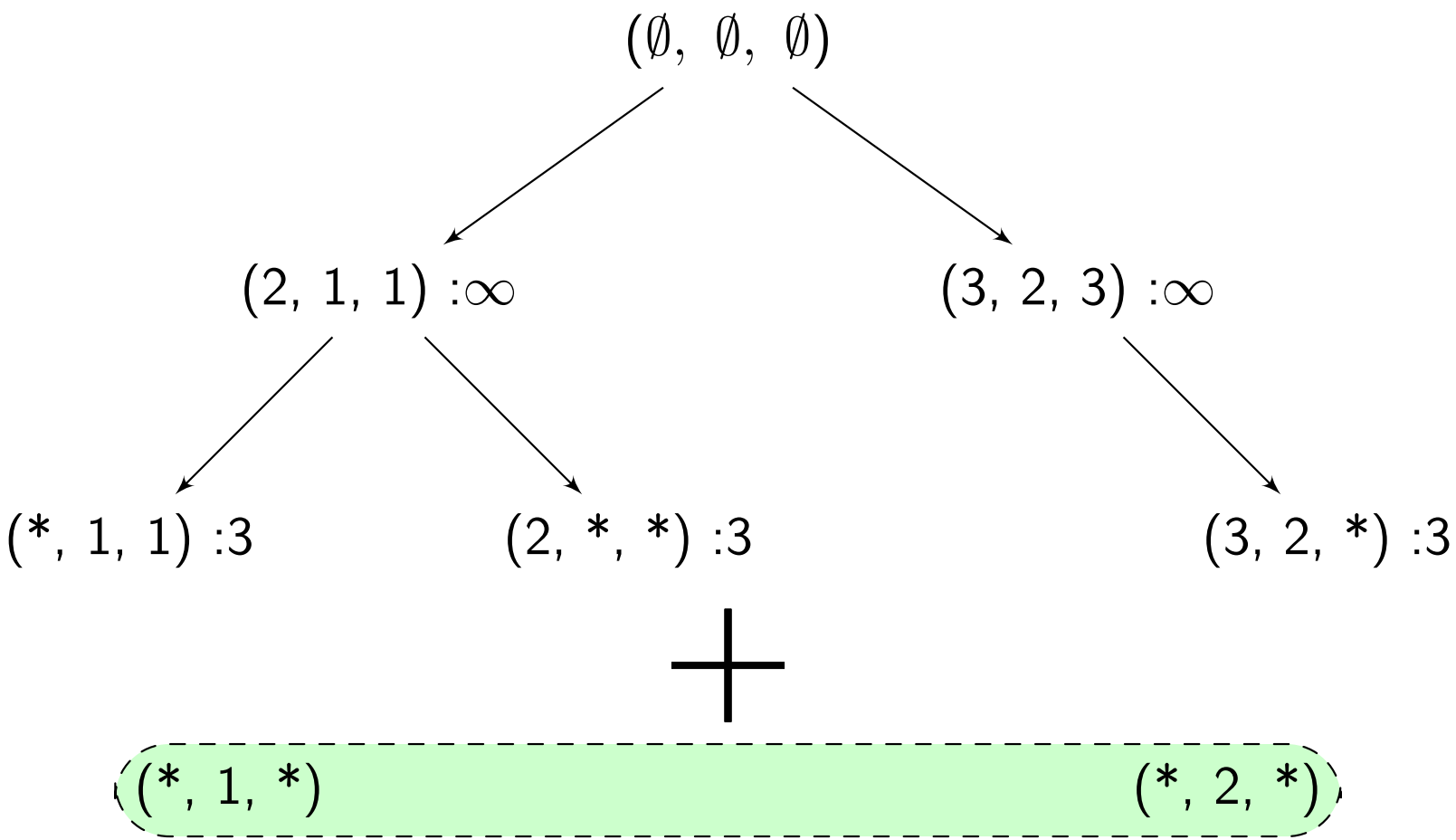
Définition ($U^\sharp$-Cube Fermé Émergent)

$$\mathbf{U^\sharp\text{-ECC}}(r_1, r_2) = \{t \in CL(r_1 \cup r_2) \mid t \text{ tuple fermé émergent}\} \cup U^\sharp$$

$U^{\sharp}$-Cube Fermé Émergent de notre exemple :



$U^\sharp$-Cube Fermé Émergent de notre exemple :



Cubes Émergents Fermés Réduits

- Pour obtenir une couverture du Cube Émergent nous avons enrichi l'ensemble des tuples fermés d'une bordure.
- Peut-on réduire l'information ajoutée aux fermés émergents tout en conservant une couverture ?
- Avant de répondre à cette question. Il faut savoir quel est le rôle exact de l'information ajoutée.

Proposition

$\forall\ t \in CL(r_1 \cup r_2)$, t est un tuple émergent si et seulement si :

$$\mathbb{C}(t, \mathbf{U^\sharp\text{-}ECC}(r_1, r_2)) \in \mathbf{U^\sharp\text{-}ECC}(r_1, r_2) \backslash U^\sharp$$

L'ajout de $U^\sharp$ permet donc de décider si un tuple t est émergent uniquement en calculant sa fermeture.

Définition (Tuple Fermé Émergent Redondant)

$\forall\ t \in\ U^\sharp$, t est un tuple fermé redondant si et seulement si $\mathbb{C}(t, \mathbf{U^\sharp\text{-}ECC}(r_1, r_2) \backslash \{t\}) = t$.

En d'autres termes un tuple de $U^\sharp$ est redondant *ssi* sa suppression du $U^\sharp$-Cube Fermé Emergent ne change pas le système de fermeture.

Cubes Émergents Fermés Réduits

Définition (Bordure Réduite $U^\sharp$)

La bordure $U^\sharp$ *Réduite*, notée $U^{\sharp\sharp}$, est la bordure $U^\sharp$ purgée de tous les tuples fermés redondants.

$$U^{\sharp\sharp} = \{t \in U^\sharp \text{ tel que } t \text{ n'est pas un tuple fermé redondant}\}$$

Définition ($U^\sharp$-Cube Fermé Émergent Réduit)

Le $U^\sharp$-Cube Fermé Émergent Réduit est défini comme suit :

$$\textbf{R-ECC}(r_1, r_2) = \{t \in CL(r_2) \mid t \text{ est un tuple émergent fermé}\} \cup U^{\sharp\sharp}$$

Cube Quotient

- Les Cubes Fermés Émergents sont des représentations du Cube Émergent qui permettent pour chaque tuple de dériver sa mesure.
- Mais les opérateurs de navigation (Roll-Up/ Drill-Down) ne sont pas préservés.
- Pour proposer une couverture du Cube Émergent préservant la sémantique de navigation nous adaptons le Cube Quotient pour le Cube Émergent en faisant le lien entre Cube Quotient et la fermeture cubique.

Définition (Relation d'équivalence quotient)

Soit f une fonction mesure. La relation d'équivalence $\equiv_f$ est dite *relation d'équivalence quotient* si et seulement si elle satisfait la propriété de congruence faible : $\forall t, t', u, u' \in CL(r)$, si $t \equiv_f t', u \equiv_f u', t \preceq_g u$ et $u' \preceq_g t'$, alors $t \equiv_f u$.

Définition (Cube Quotient)

Soit $CL(r)$ le treillis cube de la relation r et $\equiv_f$ une relation d'équivalence quotient. Le Cube Quotient de r, noté $QuotientCube(r, \equiv_f)$, est défini comme suit :

$$QuotientCube(r, \equiv_f) = \{([t]_{\equiv_f}, f_{val}(t, r)) \text{ tel que } t \in CL(r)\}.$$

Le Cube Quotient de r est une partition convexe de $CL(r)$.

Sémantique basée sur la fermeture

Définition (Relation de Couverture)

Soit $t \in CL(r)$, la couverture de t est un ensemble de tuples de r qui sont généralisés par t (*i.e.* $cov(t, r) = \{t' \in r \text{ tel que } t \preceq_g t'\}$).

Deux tuples $t, t' \in CL(r)$ sont dits équivalents selon la relation de couverture sur r, $t \equiv_{cov} t'$, s'ils ont la même couverture.

Proposition

Soit $t, t' \in CL(r)$, t est équivalent selon la couverture à t' sur r si et seulement si t et t' partagent la même fermeture cubique.

Cette proposition montre le lien fort qu'il existe entre le Cube Quotient et la fermeture cubique. Il est ainsi possible de définir un Cube Quotient à partir de l'opérateur de fermeture.

Cubes Quotient Émergent

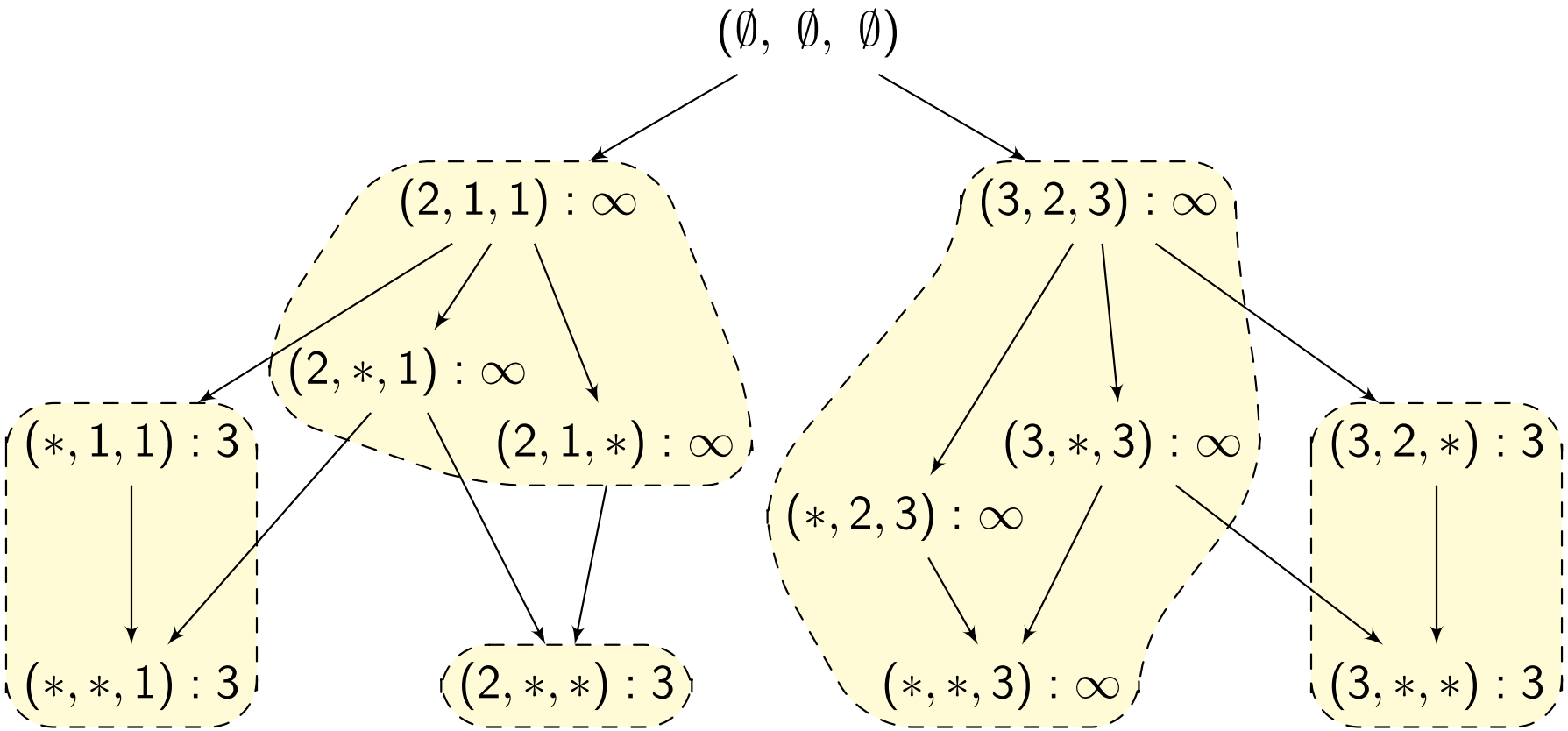
Définition (Cube Quotient Émergent)

Nous appelons Cube Quotient Émergent l'ensemble des classes d'équivalence de $CL(r_1 \cup r_2)$ émergeant de r_1 vers r_2 :
$\mathbf{EQC}(r_1, r_2) = \{([t]_{\equiv_\mathbb{C}}, ER(t))$ tel que
$[t]_{\equiv_\mathbb{C}} \in Quotient Cube(r_1 \cup r_2, \equiv_\mathbb{C})$ et t est émergent$\}$.

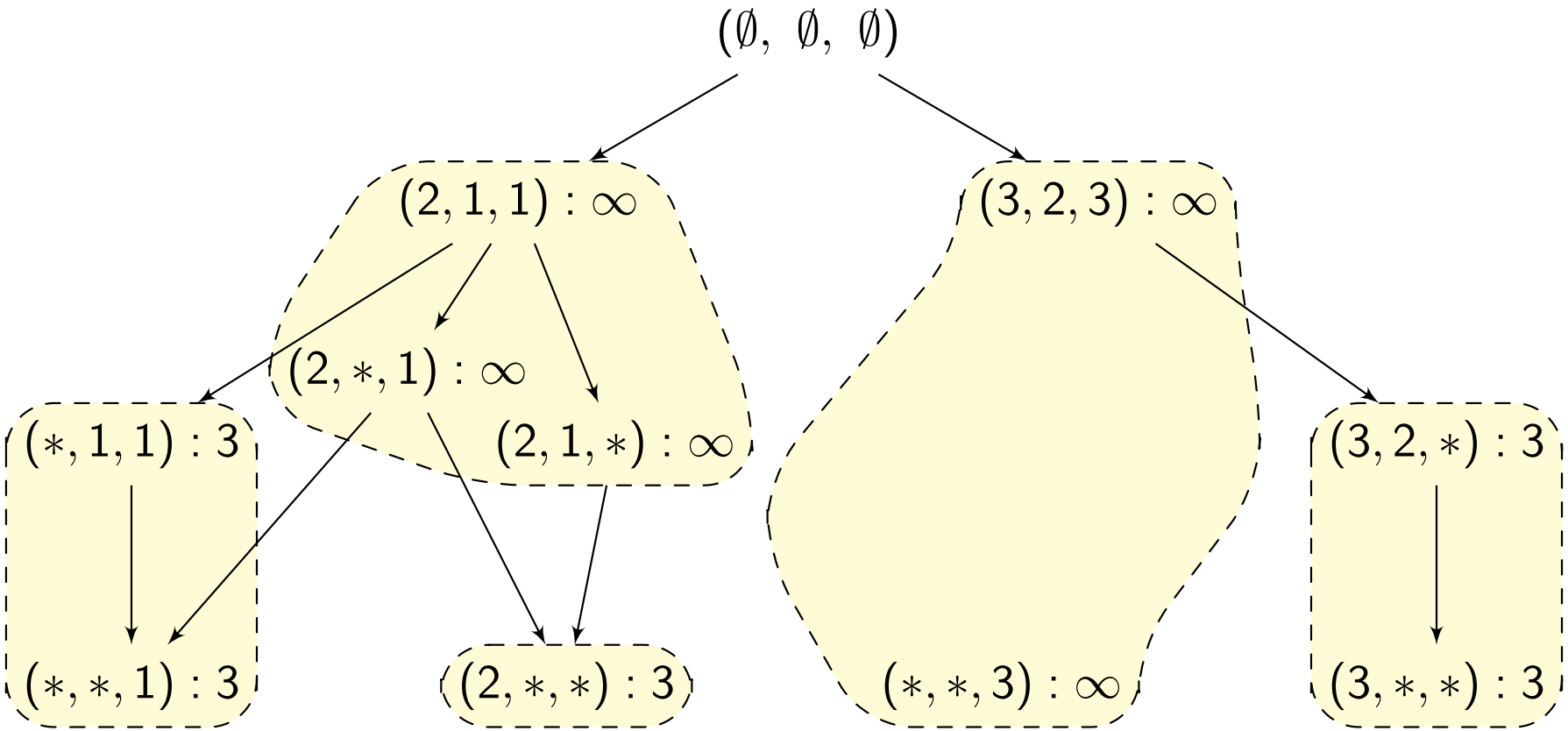
- Il est important de remarquer que les bordures $[L; U]$ sont incluses dans le Cube Quotient Émergent.
- Le lien avec la fermeture cubique garantit que pour tout tuple émergent la valeur de la mesure puisse être retrouvée.

Le Cube Quotient Émergent est donc une représentation sans perte de mesure du Cube Émergent.

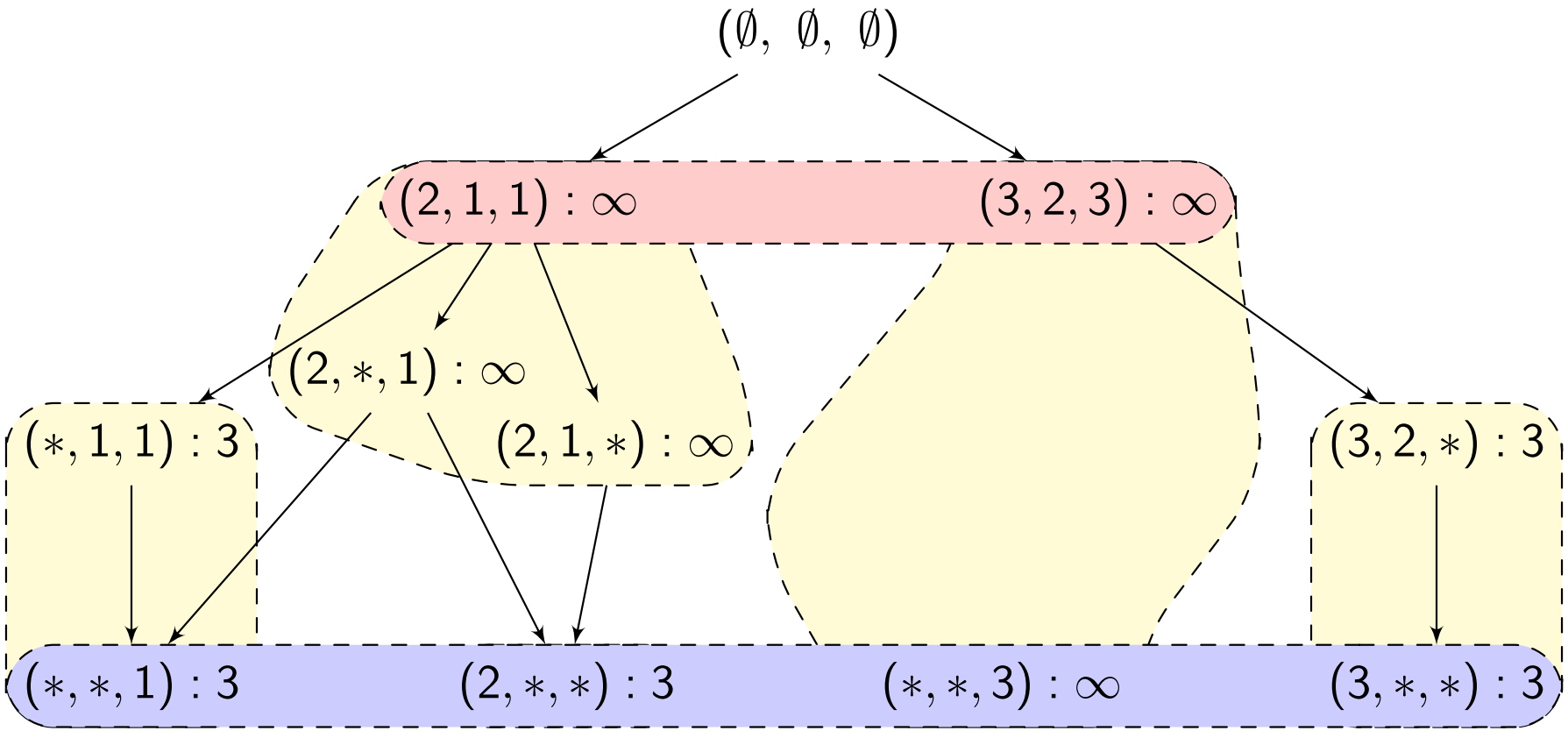
Le Cube Quotient Émergent de notre exemple :



Le Cube Quotient Émergent de notre exemple :



Le Cube Quotient Émergent de notre exemple :



Plusieurs structures ont été proposées pour représenter le Cube Émergent. Chacune a été conçue dans un objectif précis :

- Les bordures pour tester l'émergence.
- Les Cubes Fermés Émergents pour les requêtes **OLAP**.
- Le Cube Quotient Émergent pour la navigation.

Théorème

*Soit $[L; U]$, **L-ECC** and **EQC** des représentations pour le Cube Émergent (**EC**) de deux relations r_1 et r_2. Nous avons alors :*

$$[L; U] \subseteq \textbf{L-ECC} \subseteq \textbf{EQC} \subseteq \textbf{EC}$$

Ce théorème confirme l'intuition selon laquelle plus l'utilisateur a besoin de fonctionnalités, plus le volume d'informations à préserver est important.

Plusieurs structures ont été proposées pour représenter le Cube Émergent. Chacune a été conçue dans un objectif précis :

- Les bordures pour tester l'émergence.
- Les Cubes Fermés Émergents pour les requêtes **OLAP**.
- Le Cube Quotient Émergent pour la navigation.

Théorème

Soit $[L; U]$, ***L-ECC*** *and* ***EQC*** *des représentations pour le Cube Émergent (**EC**) de deux relations* r_1 *et* r_2. *Nous avons alors :*

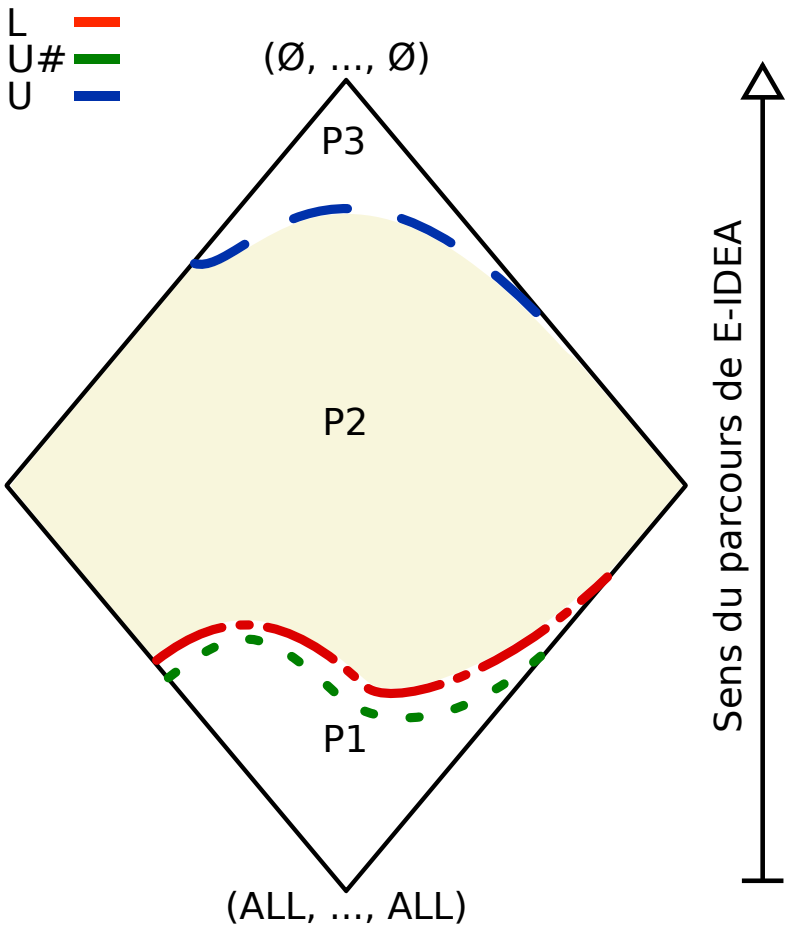
$$[L; U] \subseteq \textbf{\textit{L-ECC}} \subseteq \textbf{\textit{EQC}} \subseteq \textbf{\textit{EC}}$$

Ce théorème confirme l'intuition selon laquelle plus l'utilisateur a besoin de fonctionnalités, plus le volume d'informations à préserver est important.

Approche algorithmique

- Pour que le Cube Émergent soit un opérateur directement exploitable par l'utilisateur, les algorithmes de calcul que nous proposons sont intégrables directement dans les **SGBD**.
- Avec une approche relationnelle intégrable, il est possible de tirer pleinement partie des outils d'analyse **ROLAP** existants.
- Le Cube Émergent ne sera ainsi qu'un cube particulier et, comme on le fait déjà avec le cube de données originel, il sera possible de l'interroger, l'explorer, y naviguer.

Approche algorithmique

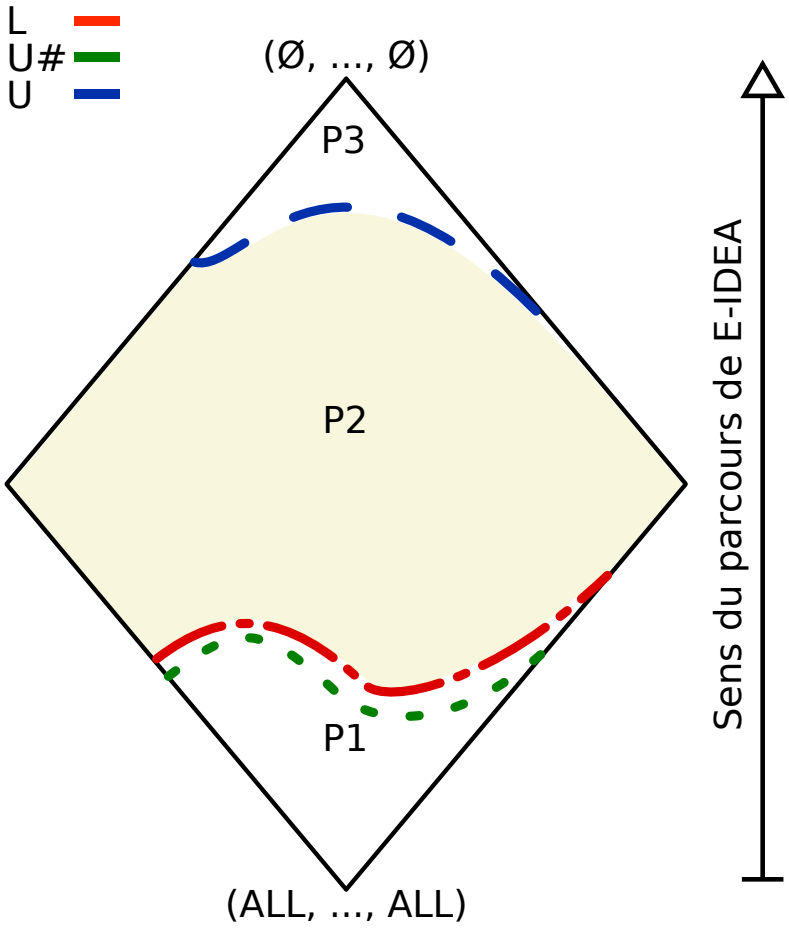


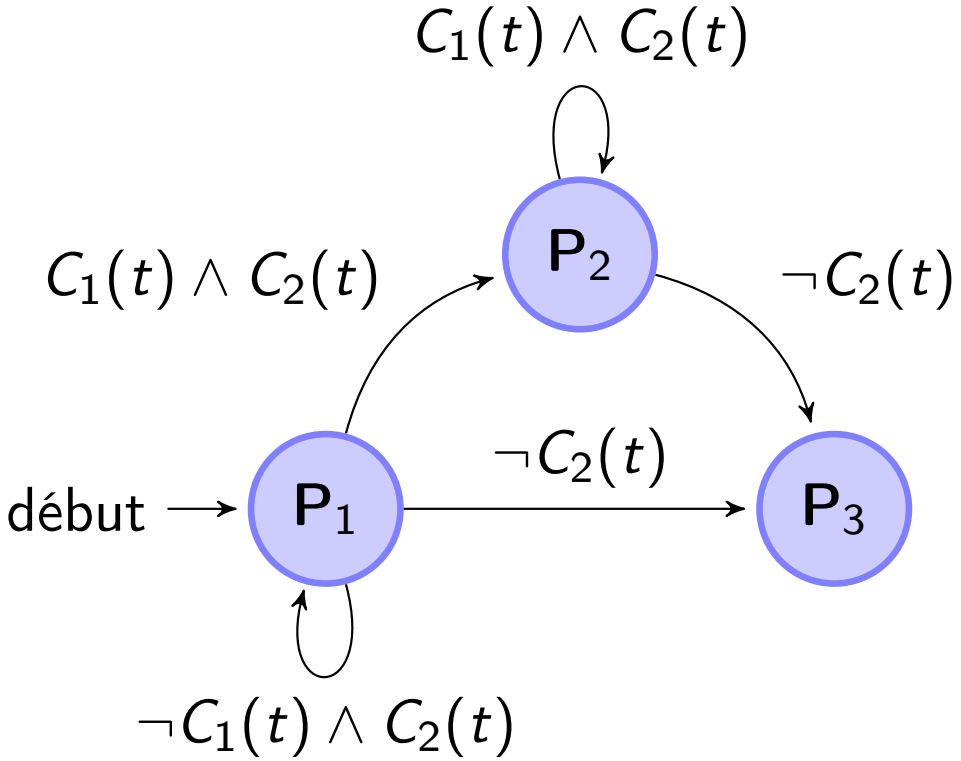
Les algorithmes parcourent le treillis cube de bas en haut et partitionne-agrége recursivement l'entrée (à la **BUC**).

Le parcours comporte trois phases :

- P_1 : tuples non émergents (le parcours doit continuer)
- P_2 : tuples émergents (phase principale)
- P_3 : tuples non émergents (arrêt du parcours)

Approche algorithmique





Plateforme IDEA (Intégrable DatabasE Algorithms)

Nous avons proposé un ensemble de concepts autour du Cube Émergent. Pour chacun de ces concepts nous avons proposé un algorithme. Tous ces algorithmes forment la plateforme logicielle IDEA.

- **E-Idea** : calcule le Cube Émergent complet.
- **F-Idea** : calcule les différentes bordures du Cube Émergent.
- $\mathbb{C}$-**Idea** : calcule les représentations du Cube Émergent basées sur la fermeture cubique.
- $\mathbb{K}$-**Idea** : calcule le Cube Quotient Émergent.
- **Idea-LogLog** : estime la taille du Cube Émergent.

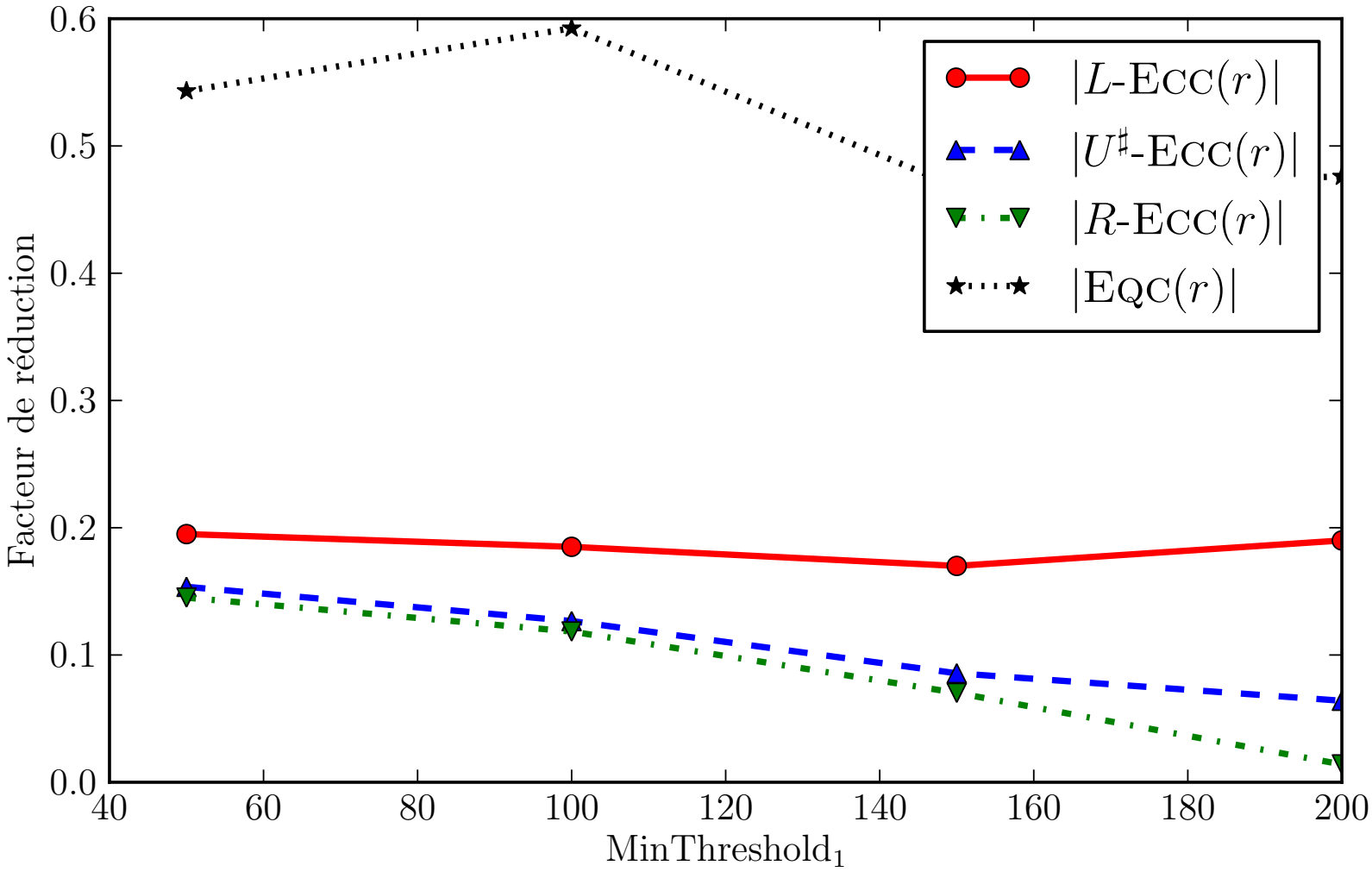
Évaluations expérimentales

Pour valider toutes nos propositions (représentations et algorithmes), nous les avons évaluées expérimentalement sur des jeux de données :

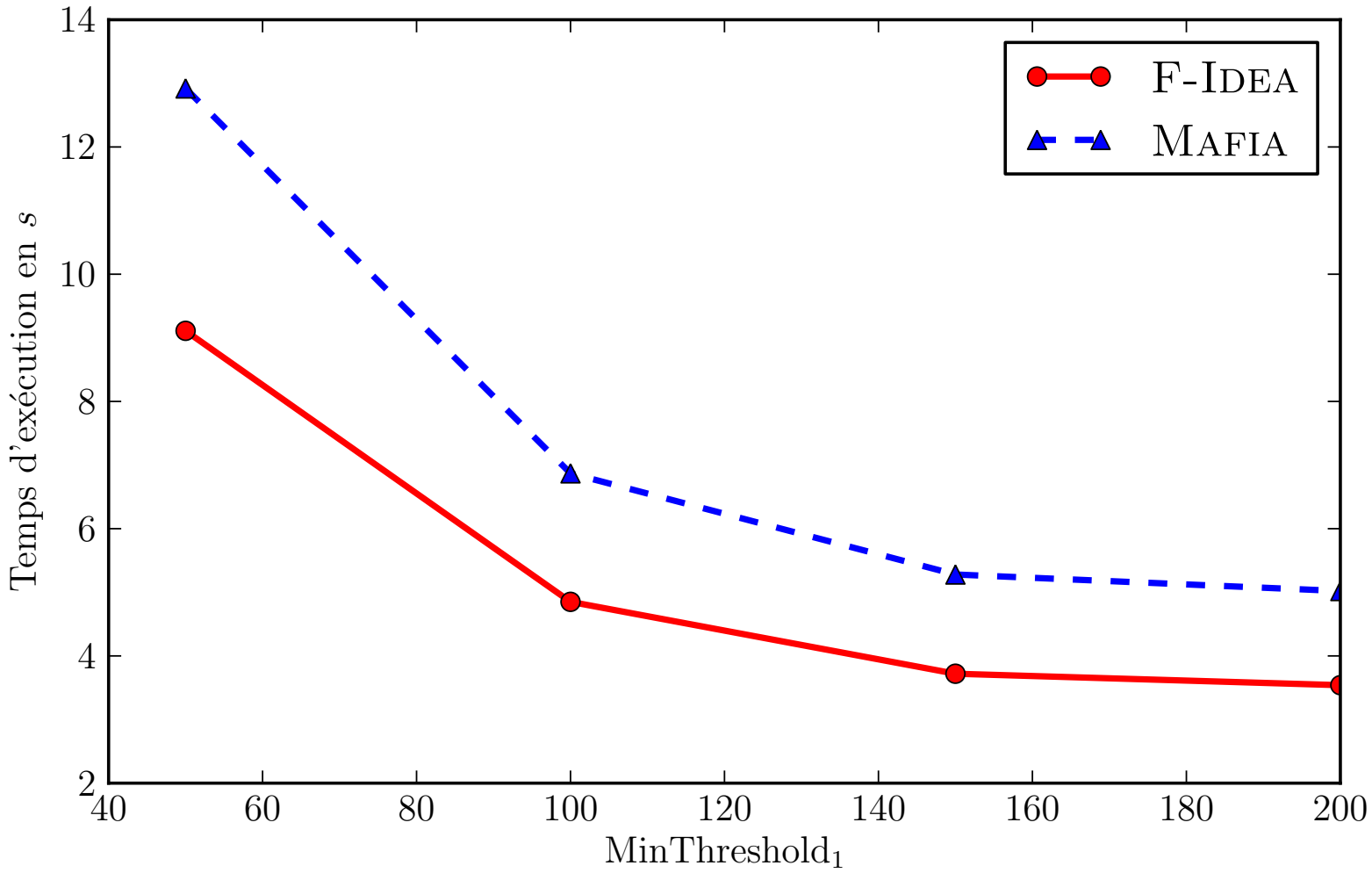
- synthétiques en modifiant plusieurs paramêtres comme le nombre de dimensions, la taille des relations, la taille des domaines des attributs dimensions ou encore le biais.
- réélles en faisant varier la contrainte d'émergence.

Les résultats obtenus sont convainquants et confirment les résultats théoriques.

Ratios de réduction pour les relations de données météorologiques :



Temps de calcul des bordures L et $U^{\sharp}$ pour les relations de données météorologiques :



Plan

- Définition d'un nouveau concept, le Cube Émergent, permettant de capturer les renversements de tendances entre plusieurs ensembles de données.
- Mise au point une méthode d'estimation de la taille pour que l'utilisateur puisse calibrer ses contraintes.
- Proposition d'un panel de représentations visant à réduire le coût de stockage en fonction des besoins des utilisateurs.
- Dévelopement d'une plateforme algorithmique intégrable au sein des **SGBD**.
- Évaluation expérimentale de touts les algorithmes et représentation.

- Évaluation des temps d'exécution des requêtes **OLAP** sur les représentations proposées et gestion du cycle de vie du Cube Émergent.
- Intégration des hiérarchies.
- Généralisation des concepts du Cube Émergent aux Cubes Contraints.
- Adaptation de $\mathbb{C}$-**Idea** pour calculer n'importe quel système de fermeture sur le treillis des parties .

Merci de votre attention.

Plan


Sameet Agarwal, Rakesh Agrawal, Prasad Deshpande, Ashish Gupta, Jeffrey F. Naughton, Raghu Ramakrishnan et Sunita Sarawagi :
On the computation of multidimensional aggregates.
In International Conference on Very Large Data Bases, pages 506–521, 1996.


Francesco Bonchi et Claudio Lucchese :
On closed constrained frequent pattern mining.
In International Conference on Data Mining, pages 35–42, 2004.


Kevin S. Beyer et Raghu Ramakrishnan :
Bottom-up computation of sparse and iceberg cubes.
In SIGMOD 1999, pages 359–370, 1999.


Alain Casali, Rosine Cicchetti et Lotfi Lakhal :
Cube lattices : A framework for multidimensional data mining.
In SIAM International Conference on Data Mining, 2003.


Alain Casali, Rosine Cicchetti et Lotfi Lakhal :
Extracting semantics from data cubes using cube transversals and closures.
In International Conference on Knowledge Discovery and Data Mining, pages 69–78, 2003.


Alain Casali, Sébastien Nedjar, Rosine Cicchetti, Lotfi Lakhal et Noel Novelli :
Lossless reduction of datacubes using partitions.
IJDWM, 5(1):18–35, 2009.


Alain Casali, Sébastien Nedjar, Rosine Cicchetti et Lotfi Lakhal :
Convex cube : Towards a unified structure for multidimensional databases.
In Database and Expert Systems Applications, pages 572–581, 2007.


Alain Casali, Sébastien Nedjar, Rosine Cicchetti et Lotfi Lakhal :
Closed cube lattices.
Annals of Information Systems, pages 145–164, 2009.


Guozhu Dong et Jinyan Li :
Mining border descriptions of emerging patterns from dataset pairs.
Knowl. Inf. Syst., 8(2):178–202, 2005.


Philippe Flajolet, Eric Fusy, Olivier Gandouet et Frédéric Meunier :
Hyperloglog : the analysis of a near-optimal cardinality estimation algorithm.
In Proceedings of the Conference on Analysis of Algorithms, AofA'07, pages 127–146, 2007.


Jim Gray, Surajit Chaudhuri, Adam Bosworth, Andrew Layman, Don Reichart, Murali Venkatrao, Frank Pellow et Hamid Pirahesh :
Data cube : A relational aggregation operator generalizing group-by, cross-tab, and sub totals.
Data Min. Knowl. Discov., 1(1):29–53, 1997.


Jiawei Han, Jian Pei, Guozhu Dong et Ke Wang :
Efficient computation of iceberg cubes with complex measures.
In SIGMOD Conference, pages 1–12, 2001.


Venky Harinarayan, Anand Rajaraman et Jeffrey D. Ullman :
Implementing data cubes efficiently.
In International Conference on Management of Data, pages 205–216, 1996.


Marc Laporte, Noel Novelli, Rosine Cicchetti et Lotfi Lakhal :
Computing full and iceberg datacubes using partitions.
In ISMIS, pages 244–254, 2002.


Laks V. S. Lakshmanan, Jian Pei et Jiawei Han :
Quotient cube : How to summarize the semantics of a data cube.
In International Conference on Very Large Data Bases, pages 778–789, 2002.


Konstantinos Morfonios et Yannis E. Ioannidis :
Supporting the data cube lifecycle : the power of rolap.
VLDB J., 17(4):729–764, 2008.


Sébastien Nedjar, Alain Casali, Rosine Cicchetti et Lotfi Lakhal :
Cocktail de cubes.
In Dominique Laurent, éditeur : *22èmes Journées Bases de Données Avancées*, 2006.


Sébastien Nedjar, Alain Casali, Rosine Cicchetti et Lotfi Lakhal :
Cubes convexes.
Ingénierie des Systèmes d'Information, 11(6):11–31, 2006.


Sébastien Nedjar, Alain Casali, Rosine Cicchetti et Lotfi Lakhal :
Emerging cubes for trends analysis in olap databases.
In Data Warehousing and Knowledge Discovery. Springer, 2007.


Sébastien Nedjar, Alain Casali, Rosine Cicchetti et Lotfi Lakhal :
Résumés du cube emergent.
In 23èmes Journées Bases de Données Avancées,, 2007.


Sébastien Nedjar, Alain Casali, Rosine Cicchetti et Lotfi Lakhal :
Upper borders for emerging cubes.
In Data Warehousing and Knowledge Discovery, pages 45–54, 2008.


Sébastien Nedjar, Alain Casali, Rosine Cicchetti et Lotfi Lakhal :
Emerging cubes : Borders, size estimations and lossless reductions.
Information Systems, 34(6):536–550, 2009.


Sébastien Nedjar, Alain Casali, Rosine Cicchetti et Lotfi Lakhal :
Cubes fermés / quotients Émergents.
In Extraction et gestion des connaissances, Revue des Nouvelles Technologies de l'Information, 2010.
à paraitre.


Sébastien Nedjar, Alain Casali, Rosine Cicchetti et Lotfi Lakhal :
Reduced representations of emerging cubes for olap database mining.
IJBIDM, 5(1):268–300, 2010.


Sébastien Nedjar :
Exact and approximate sizes of convex datacubes.
In Data Warehousing and Knowledge Discovery, pages 204–215, 2009.


Dong Xin, Jiawei Han, Xiaolei Li, Zheng Shao et Benjamin W. Wah :
Computing iceberg cubes by top-down and bottom-up integration : The starcubing approach.
IEEE Trans. Knowl. Data Eng., 19(1):111–126, 2007.

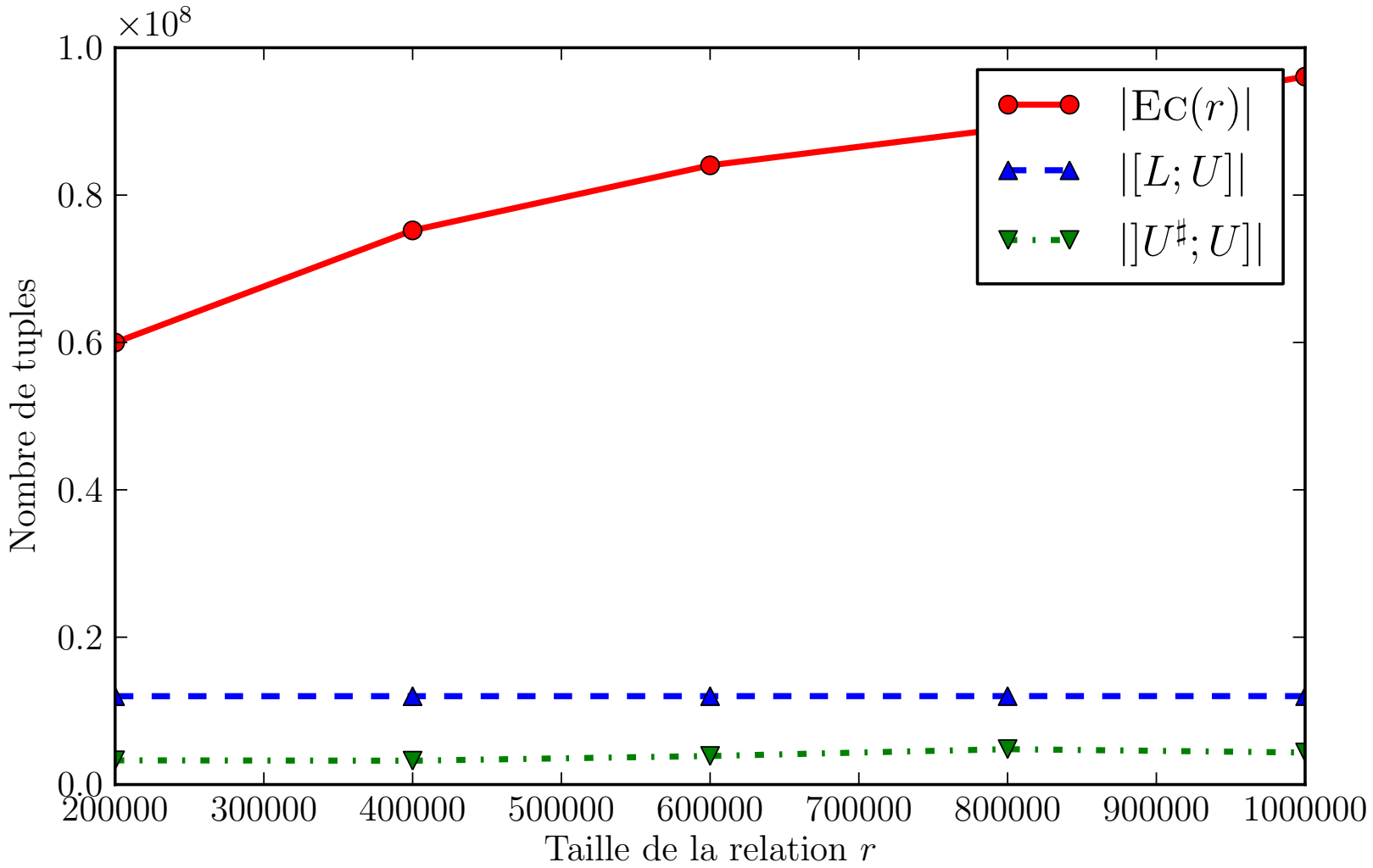

Guizhen Yang :
The complexity of mining maximal frequent itemsets and maximal frequent patterns.
In International Conference on Knowledge Discovery and Data Mining, pages 344–353, 2004.


Guizhen Yang :
Computational aspects of mining maximal frequent patterns.
Theor. Comput. Sci., 362(1-3):63–85, 2006.

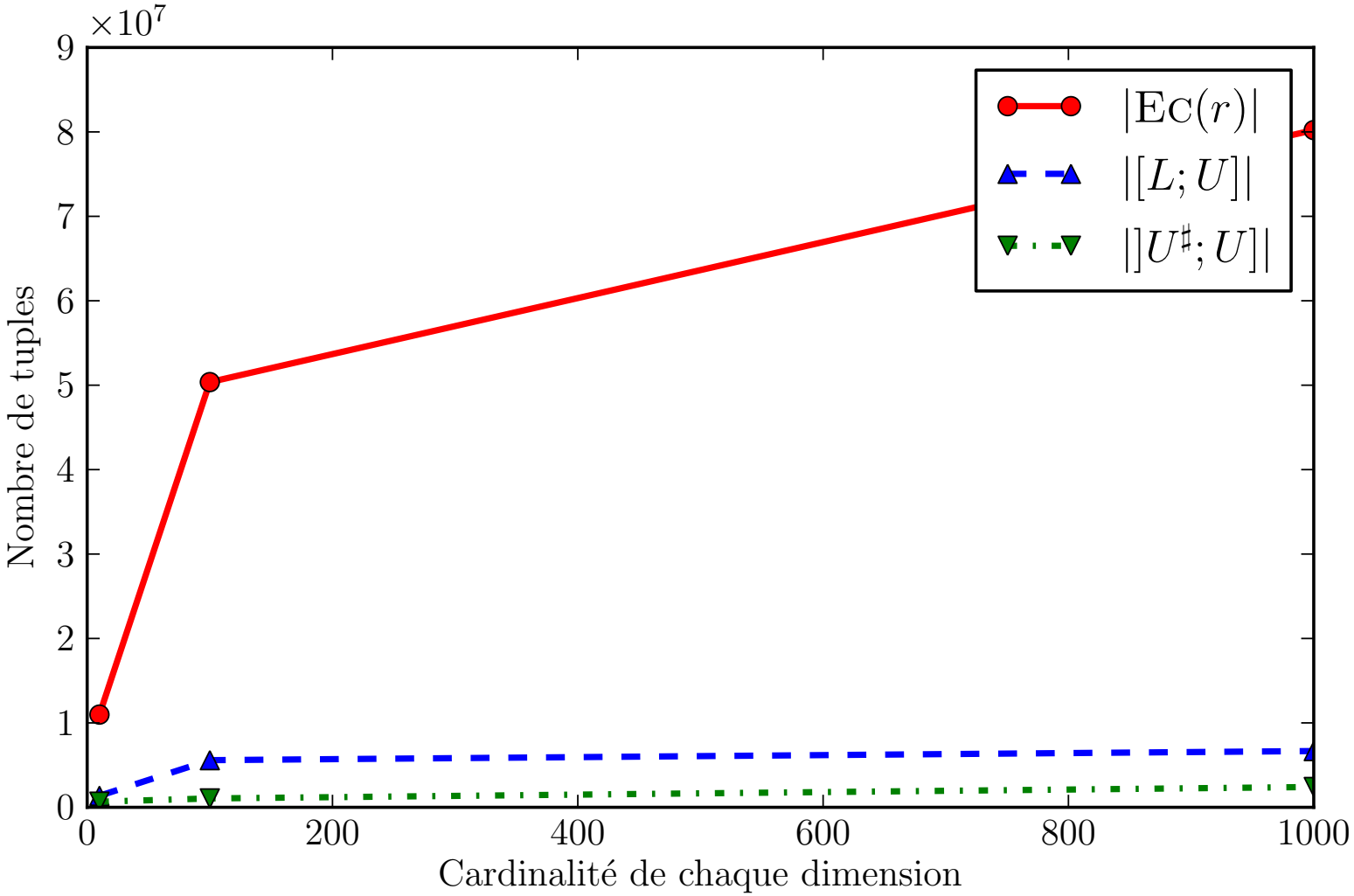

Xiuzhen Zhang, Pauline Lienhua Chou et Guozhu Dong :
Efficient computation of iceberg cubes by bounding aggregate functions.
IEEE Trans. Knowl. Data Eng., 19(7):903–918, 2007.

Plan

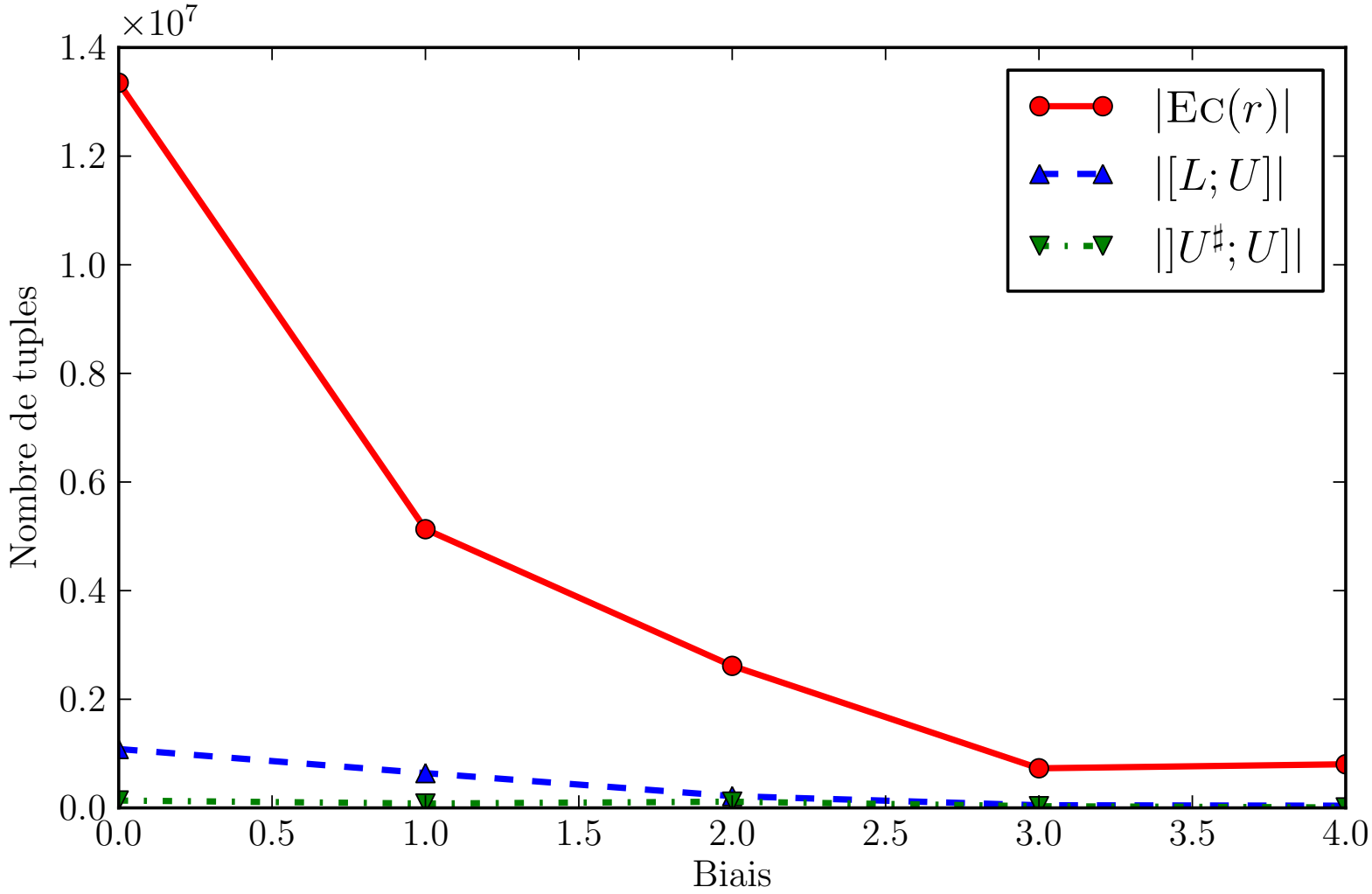
Taille du Cube Émergent et de ses bordures $[L; U]$ et $]U^\sharp; U]$ avec $\mathcal{D} = 10$, $\mathcal{C} = 100$, $\mathcal{S} = 0$



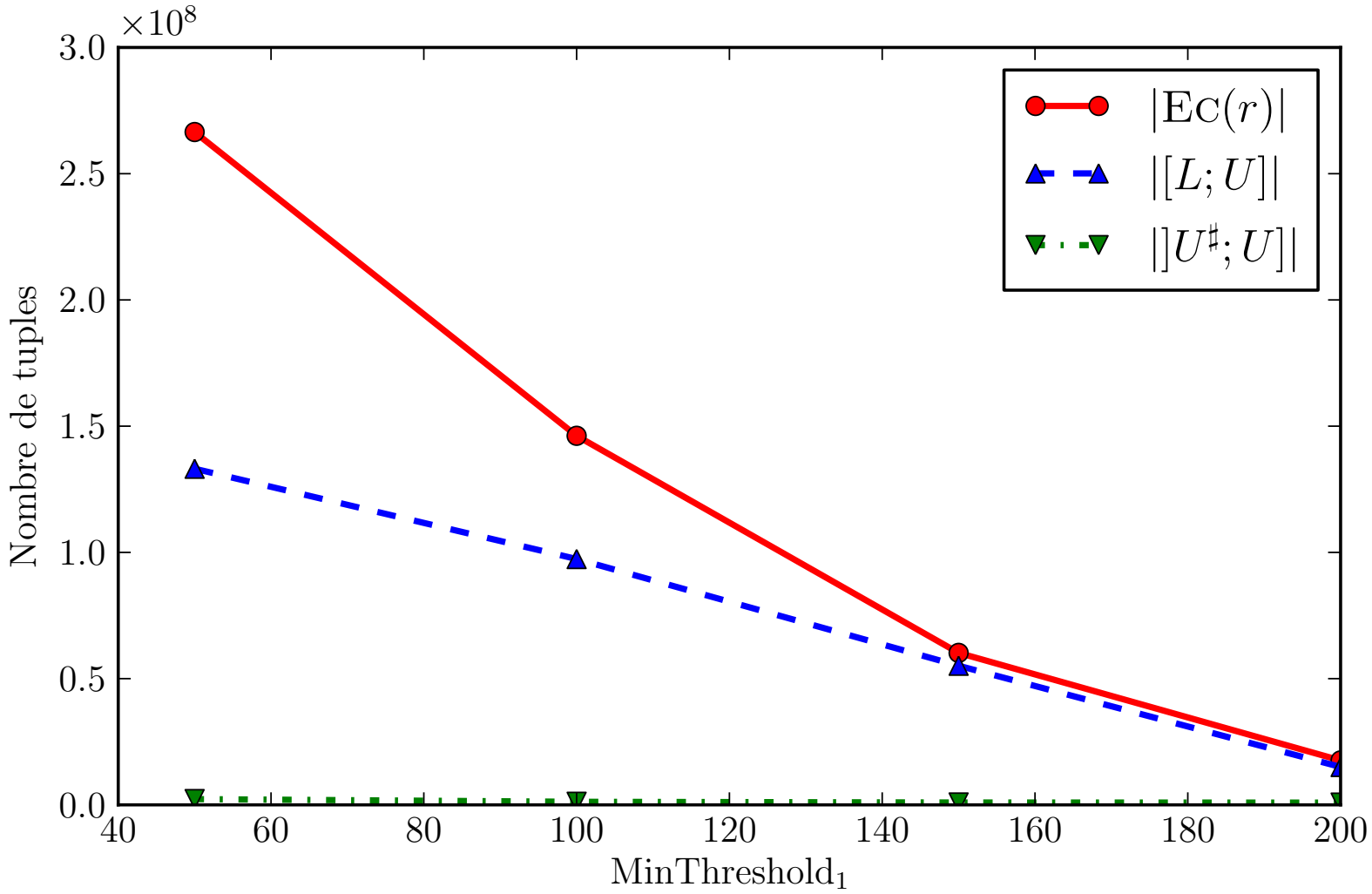
Taille du Cube Émergent et de ses bordures $[L; U]$ et $]U^\sharp; U]$ avec $\mathcal{D} = 10$, $\mathcal{T} = 1000\text{K}$, $\mathcal{S} = 0$



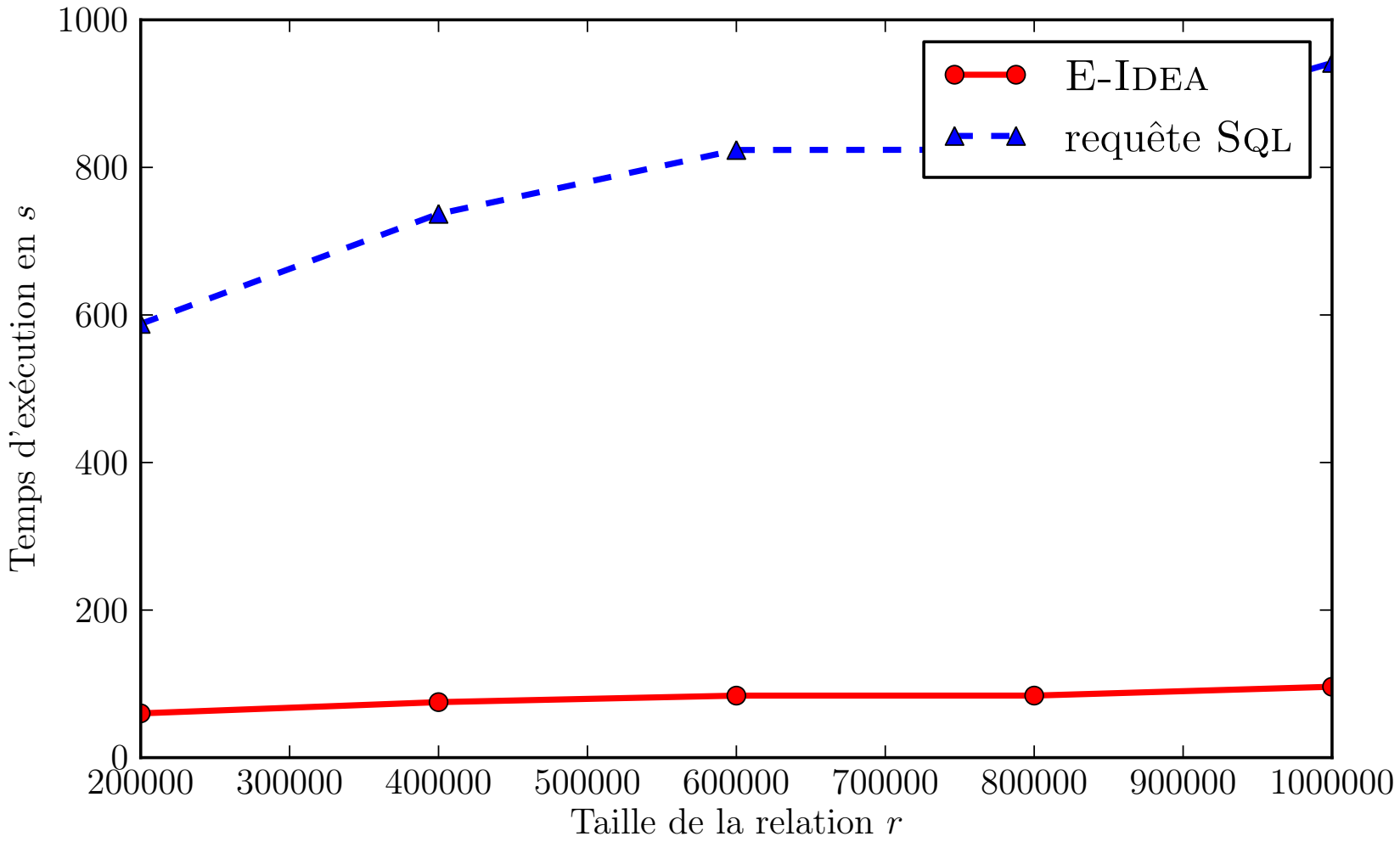
Taille du Cube Émergent et de ses bordures $[L; U]$ et $]U^\sharp; U]$ avec $\mathcal{D} = 10$, $\mathcal{C} = 100$, $\mathcal{T} = 1000$K



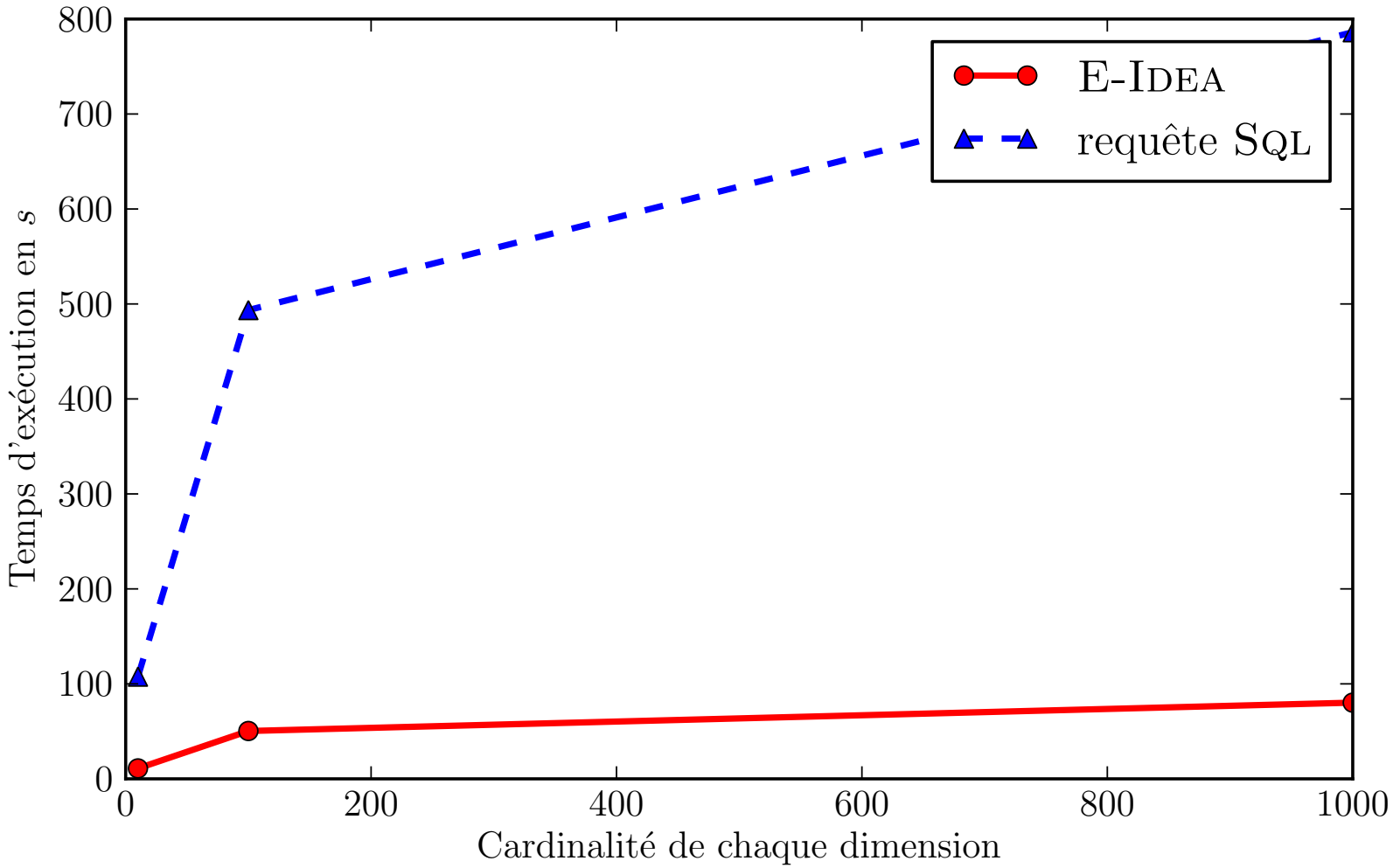
Taille du Cube Émergent et de ses bordures $[L; U]$ et $]U^{\sharp}; U]$ pour les relations de données météorologiques



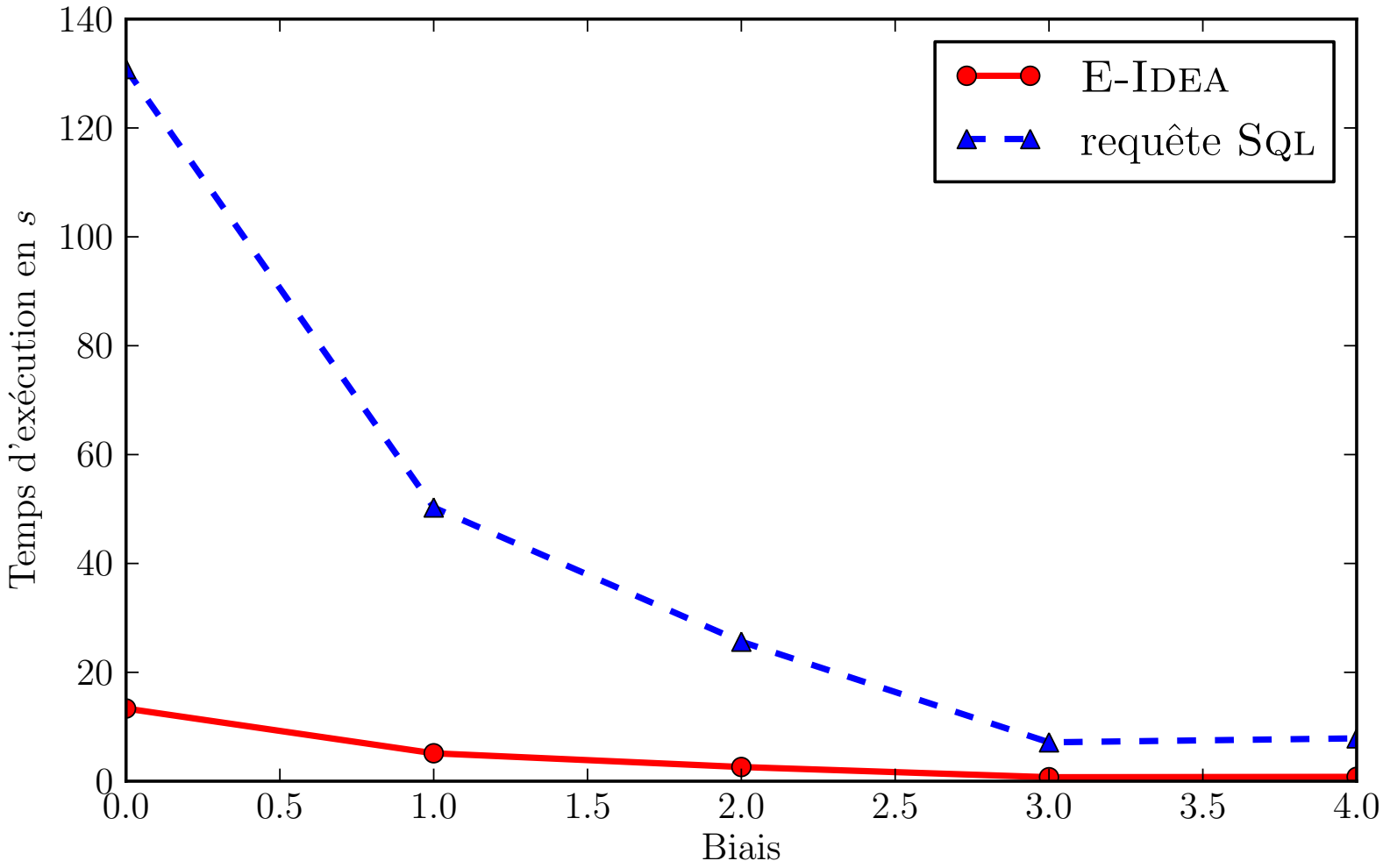
Temps de calcul du Cube Émergent avec $\mathcal{D} = 10$, $\mathcal{C} = 100$, $\mathcal{S} = 0$



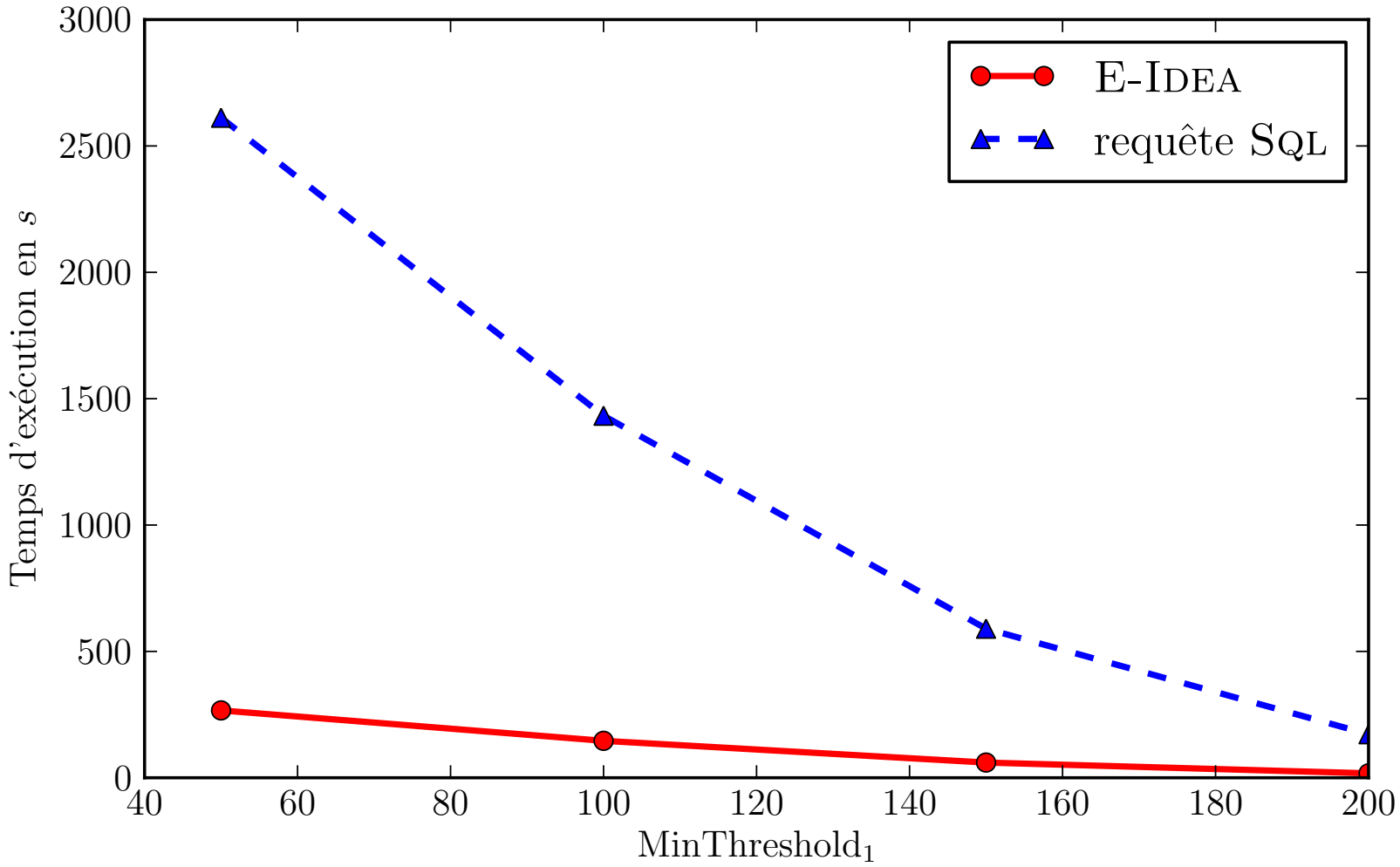
Temps de calcul du Cube Émergent avec $\mathcal{D} = 10$, $\mathcal{T} = 1000$K, $\mathcal{S} = 0$



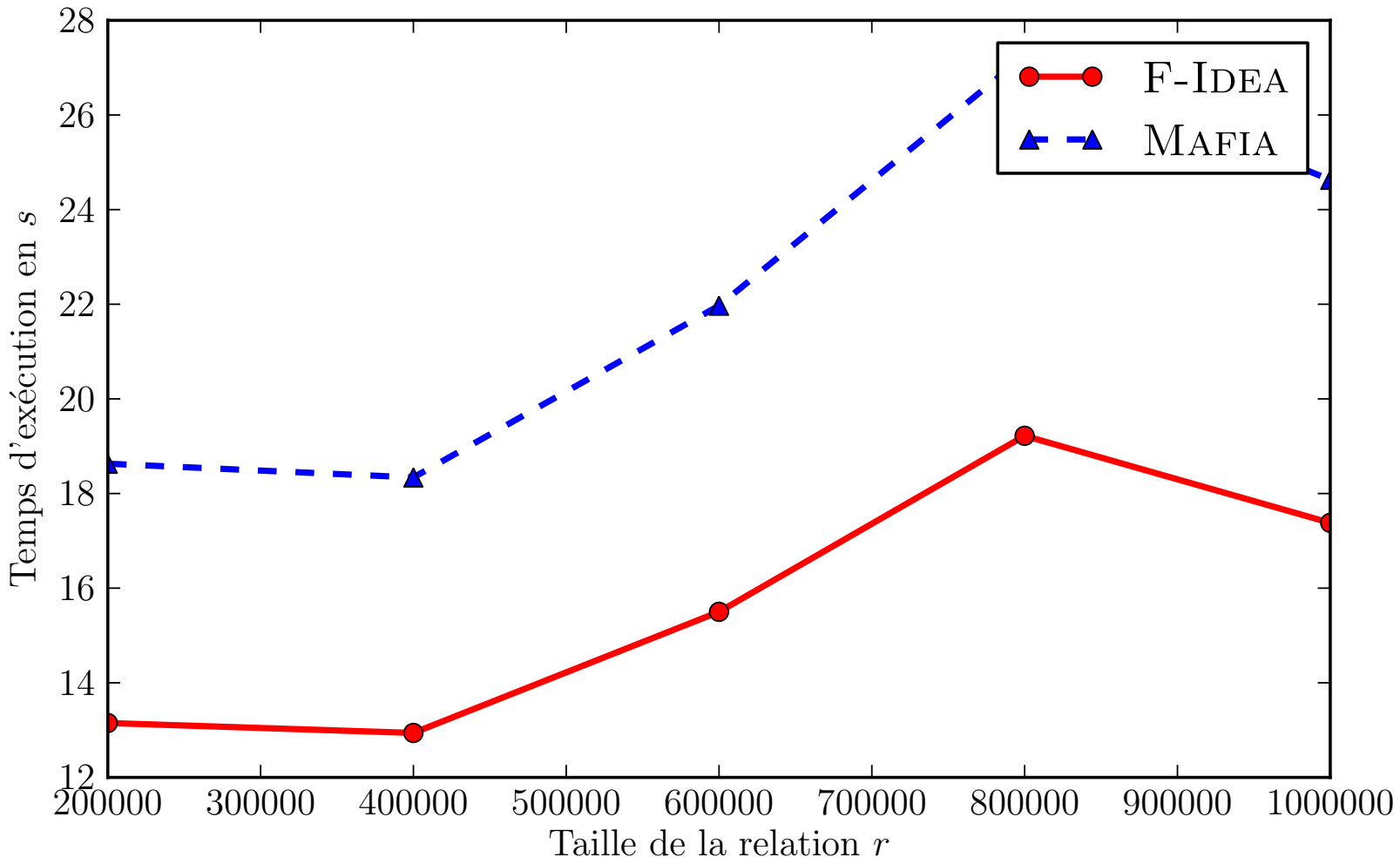
Temps de calcul du Cube Émergent avec $\mathcal{D} = 10$, $\mathcal{C} = 100$, $\mathcal{T} = 1000K$



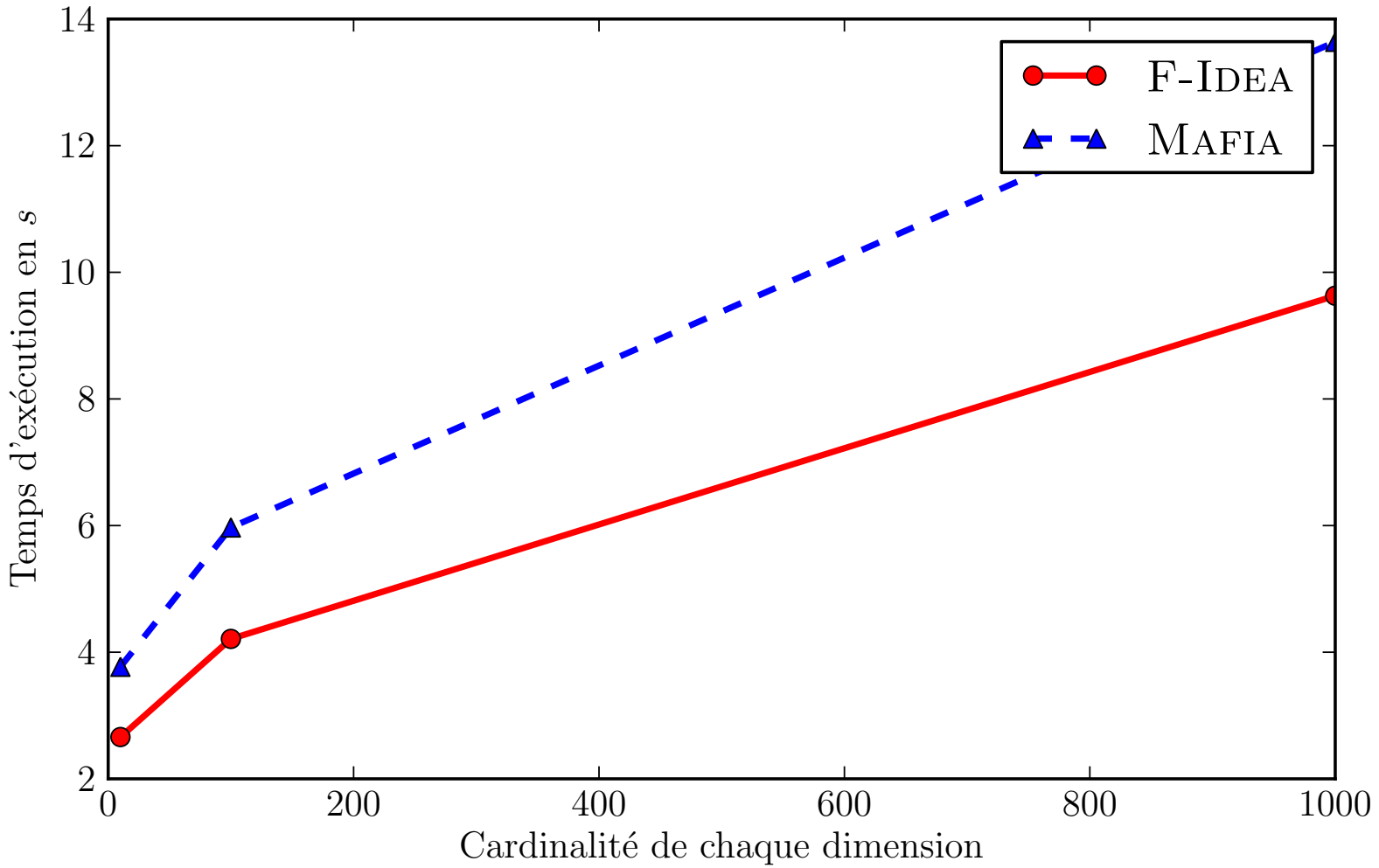
Temps de calcul du Cube Émergent pour les relations de données météorologiques



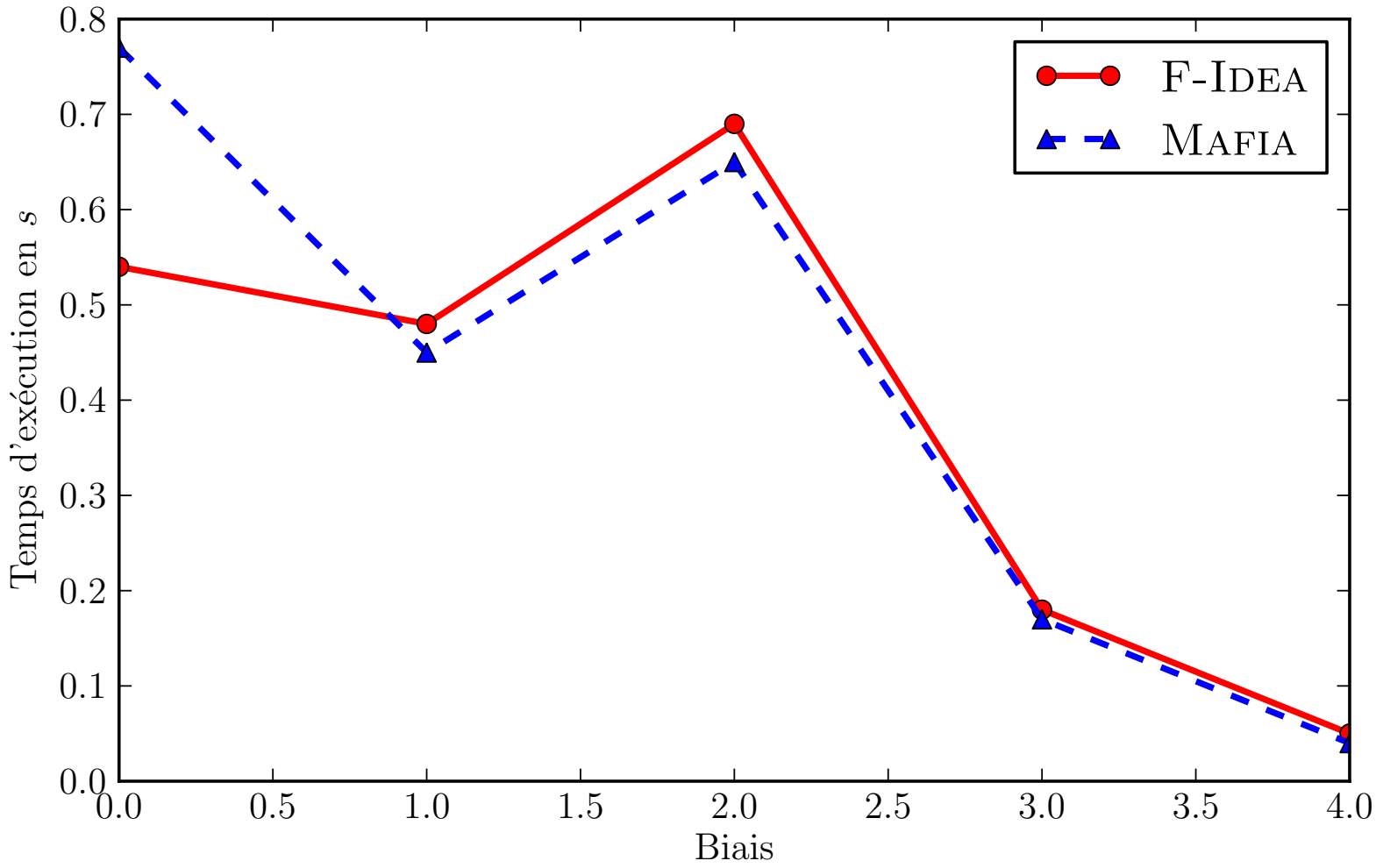
Temps de calcul des bordures L et $U^\sharp$ avec $\mathcal{D} = 10$, $\mathcal{C} = 100$, $\mathcal{S} = 0$



Temps de calcul des bordures L et $U^\sharp$ avec $\mathcal{D} = 10$, $\mathcal{T} = 1000\text{K}$, $\mathcal{S} = 0$

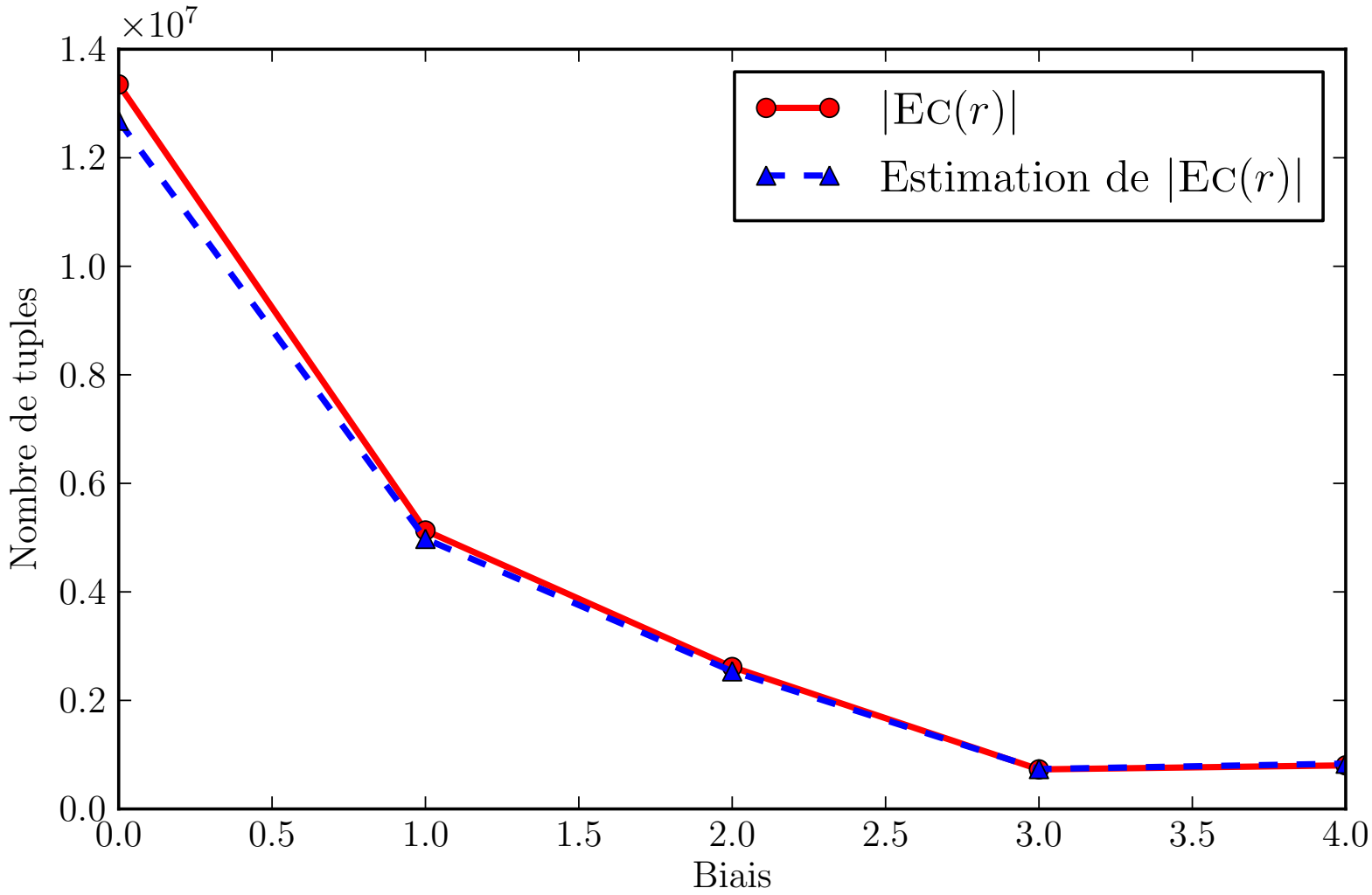


Temps de calcul des bordures L et $U^\sharp$ avec $\mathcal{D} = 10$, $\mathcal{C} = 100$, $\mathcal{T} = 1000$K

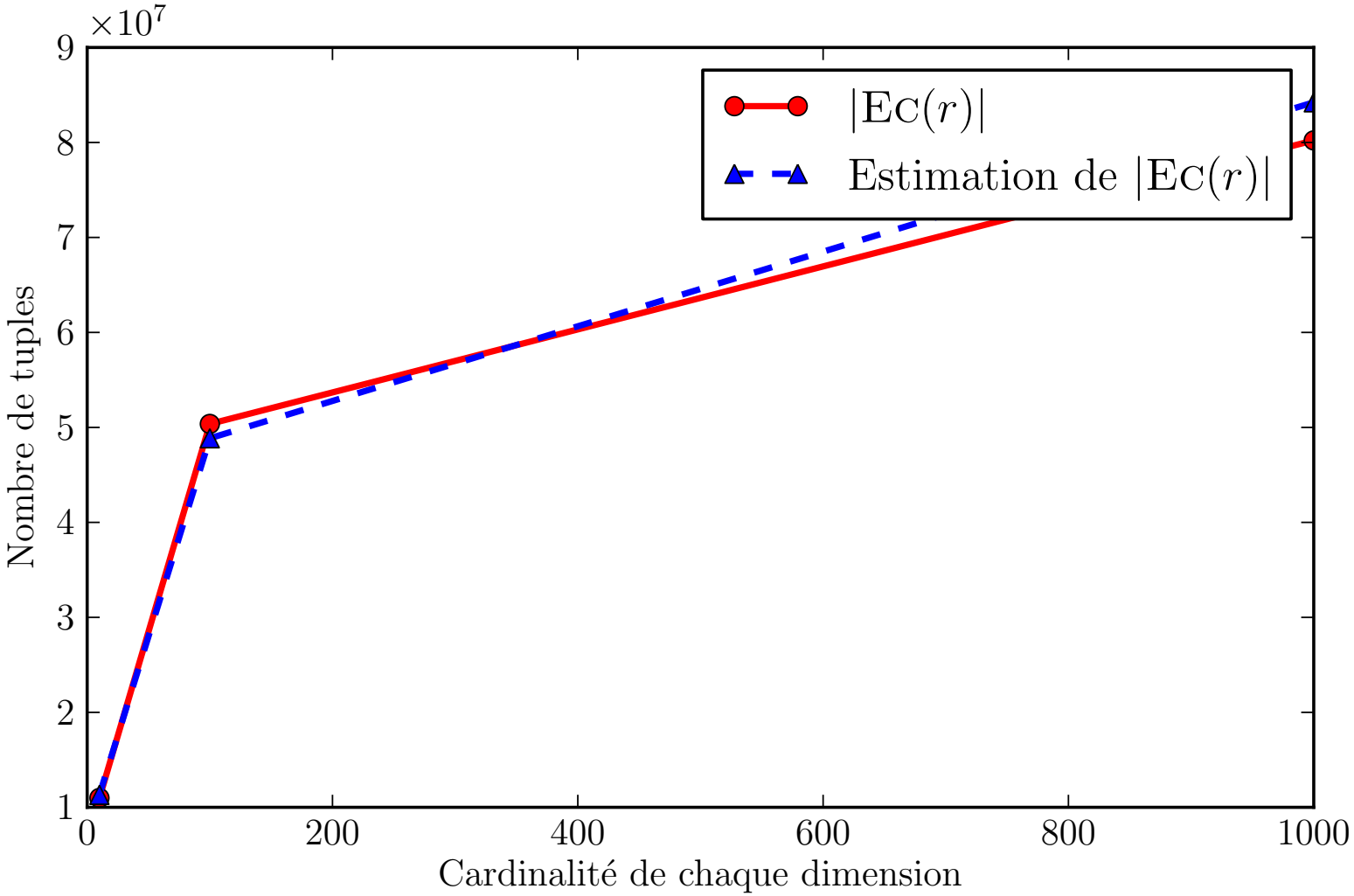


Plan

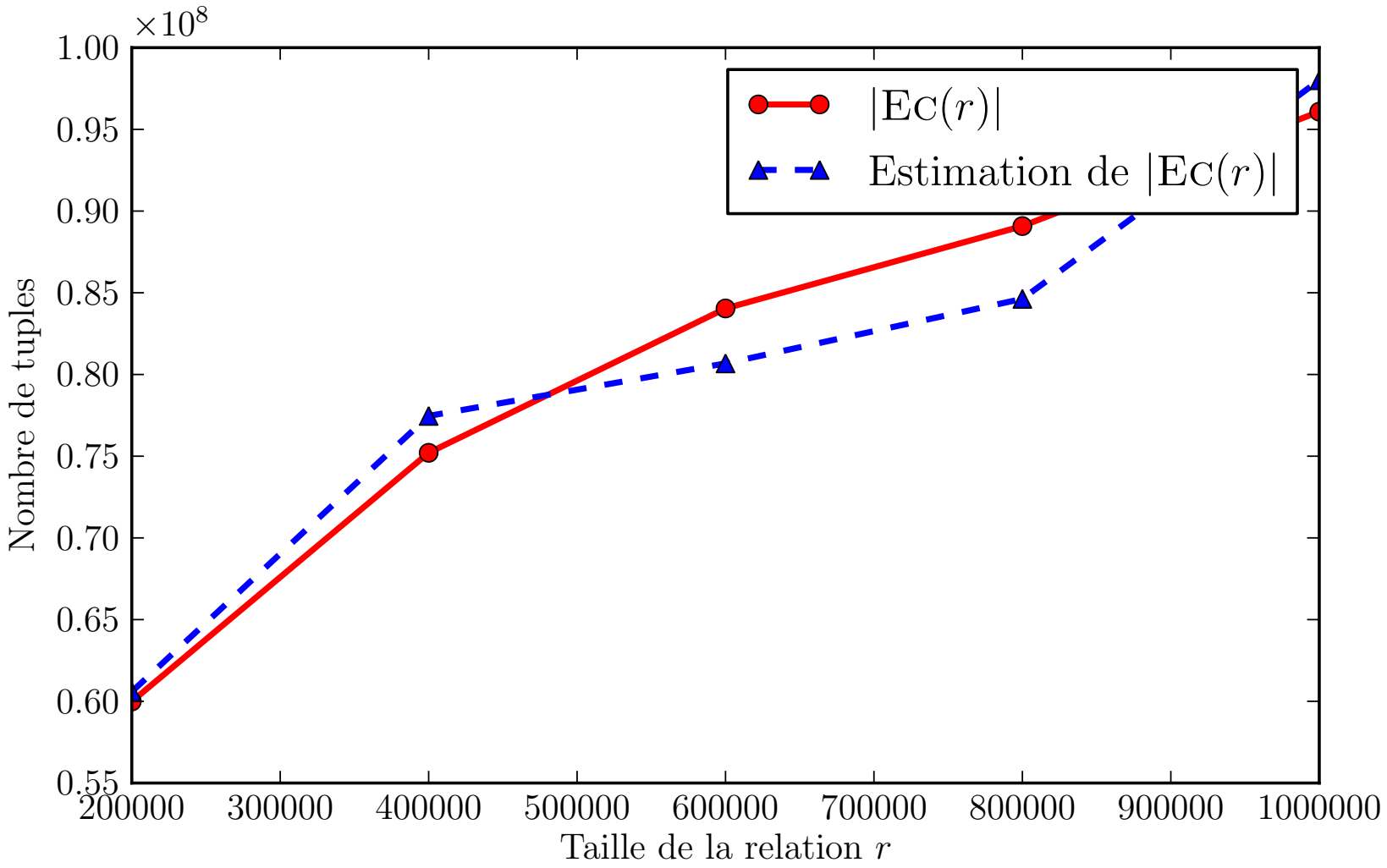
Tailles exacte et approximative du Cube Émergent avec $\mathcal{D}$=10, $\mathcal{C}$=100, $\mathcal{T}$=1000K :



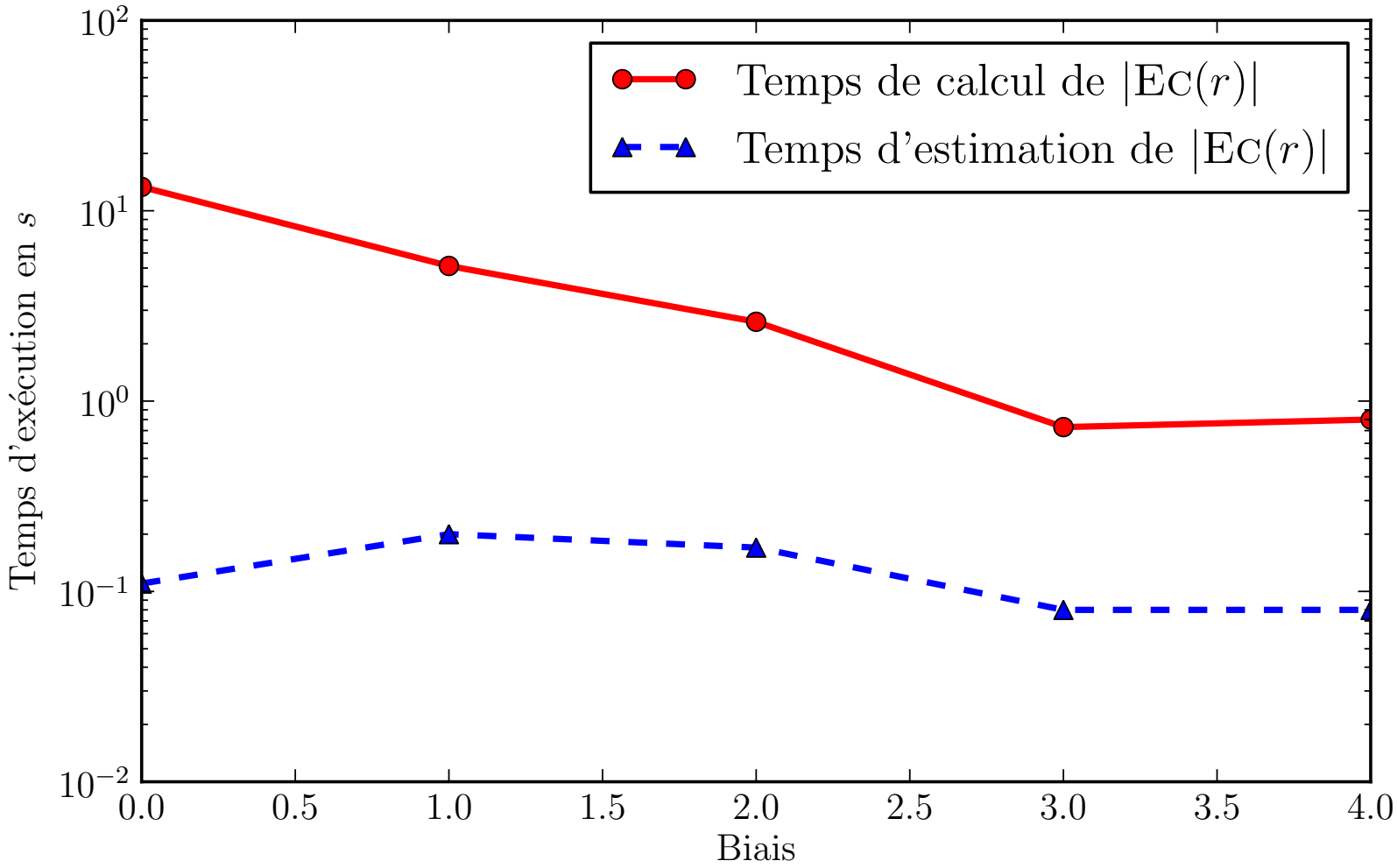
Tailles exacte et approximative du Cube Émergent avec $\mathcal{D}$=10, $\mathcal{T}$=1000K, $\mathcal{S}$=0 :



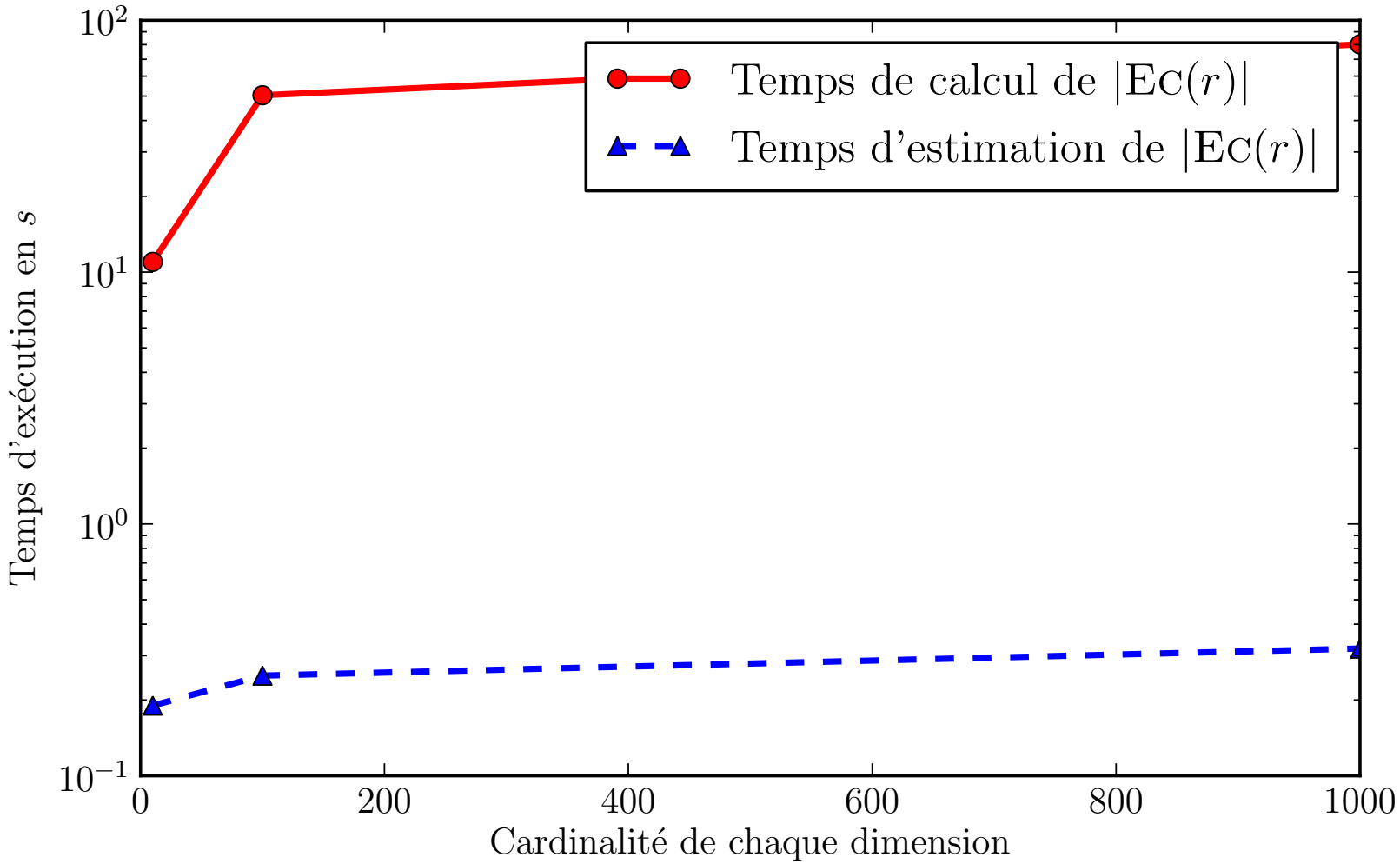
Tailles exacte et approximative du Cube Émergent avec $\mathcal{D}$=10, $\mathcal{C}$=100, $\mathcal{S}$=0 :



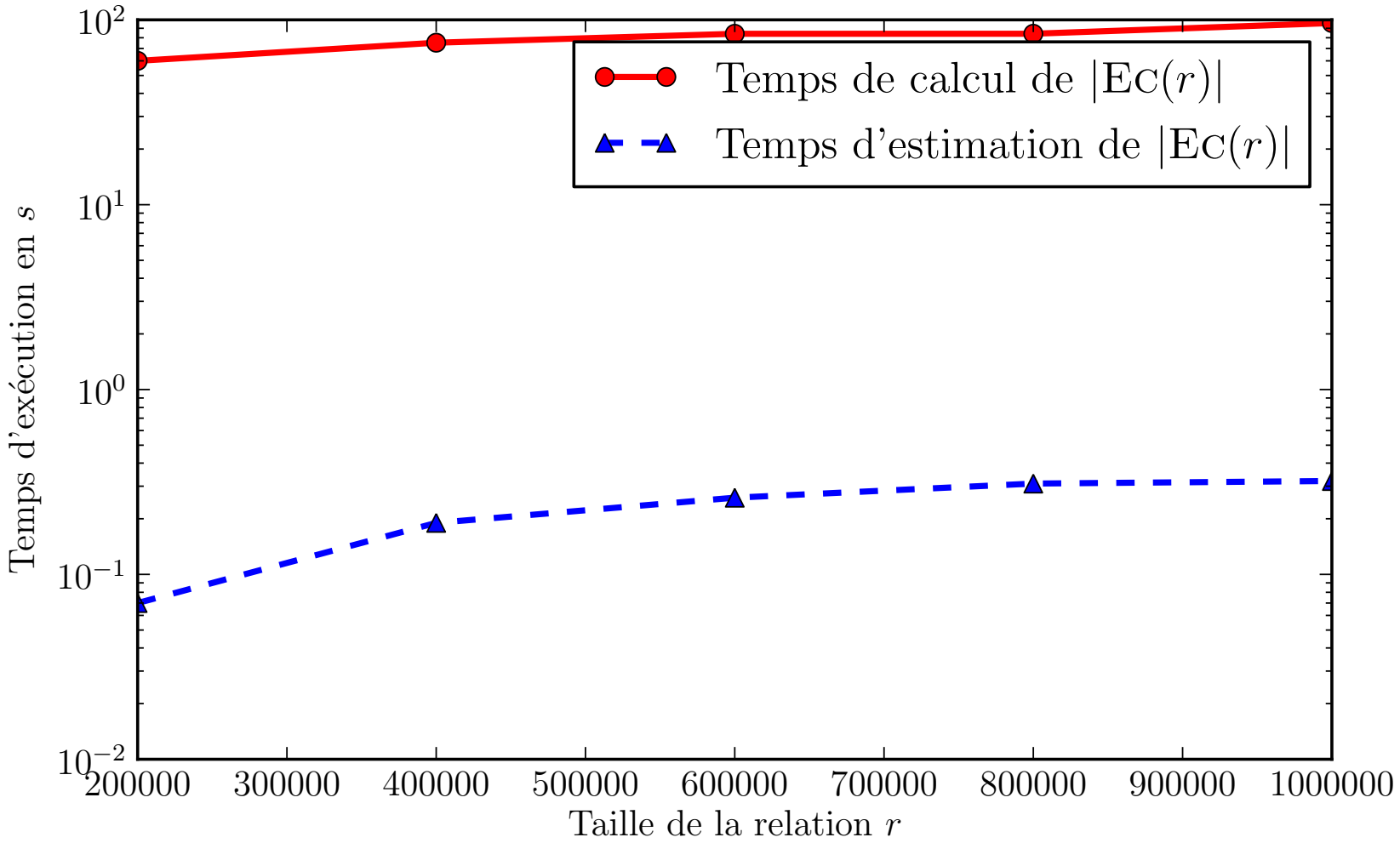
Estimation et temps de calcul du Cube Émergent avec $\mathcal{D}$=10, $\mathcal{C}$=100, $\mathcal{T}$=1000K :



Estimation et temps de calcul du Cube Émergent avec $\mathcal{D}$=10, $\mathcal{T}$=1000K, $\mathcal{S}$=0 :

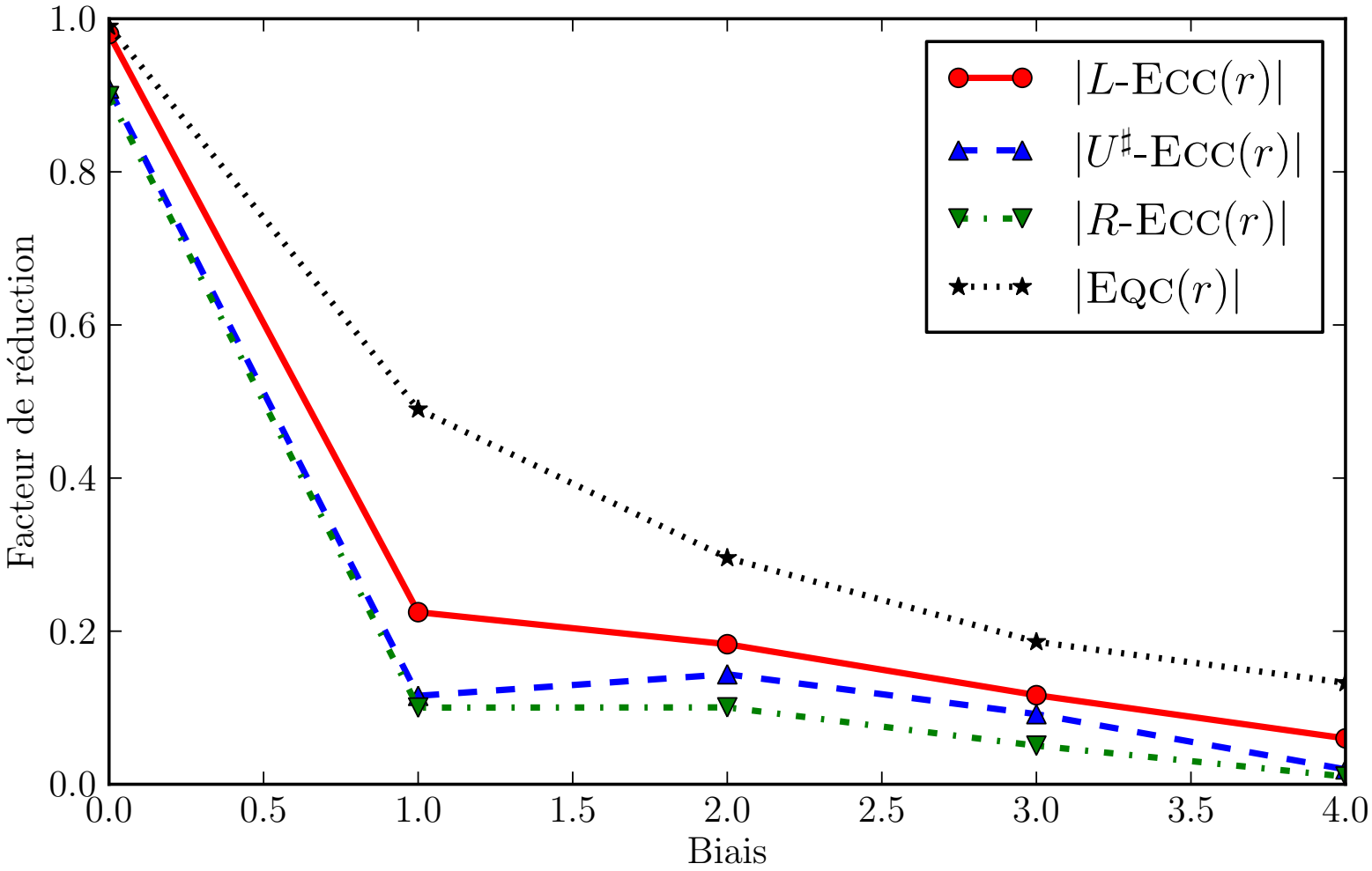


Estimation et temps de calcul du Cube Émergent avec $\mathcal{D}$=10, $\mathcal{C}$=100, $\mathcal{S}$=0 :

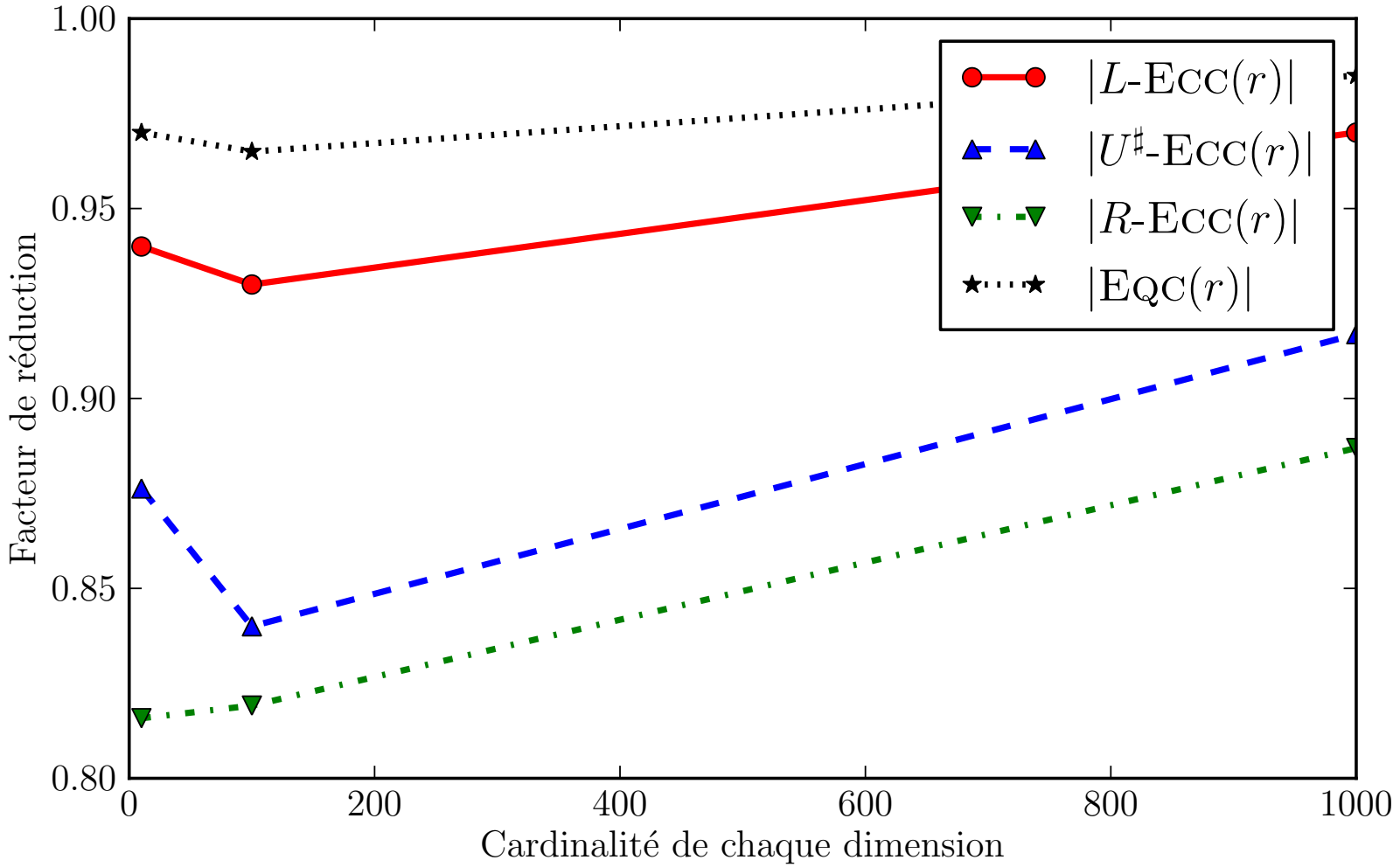


Plan

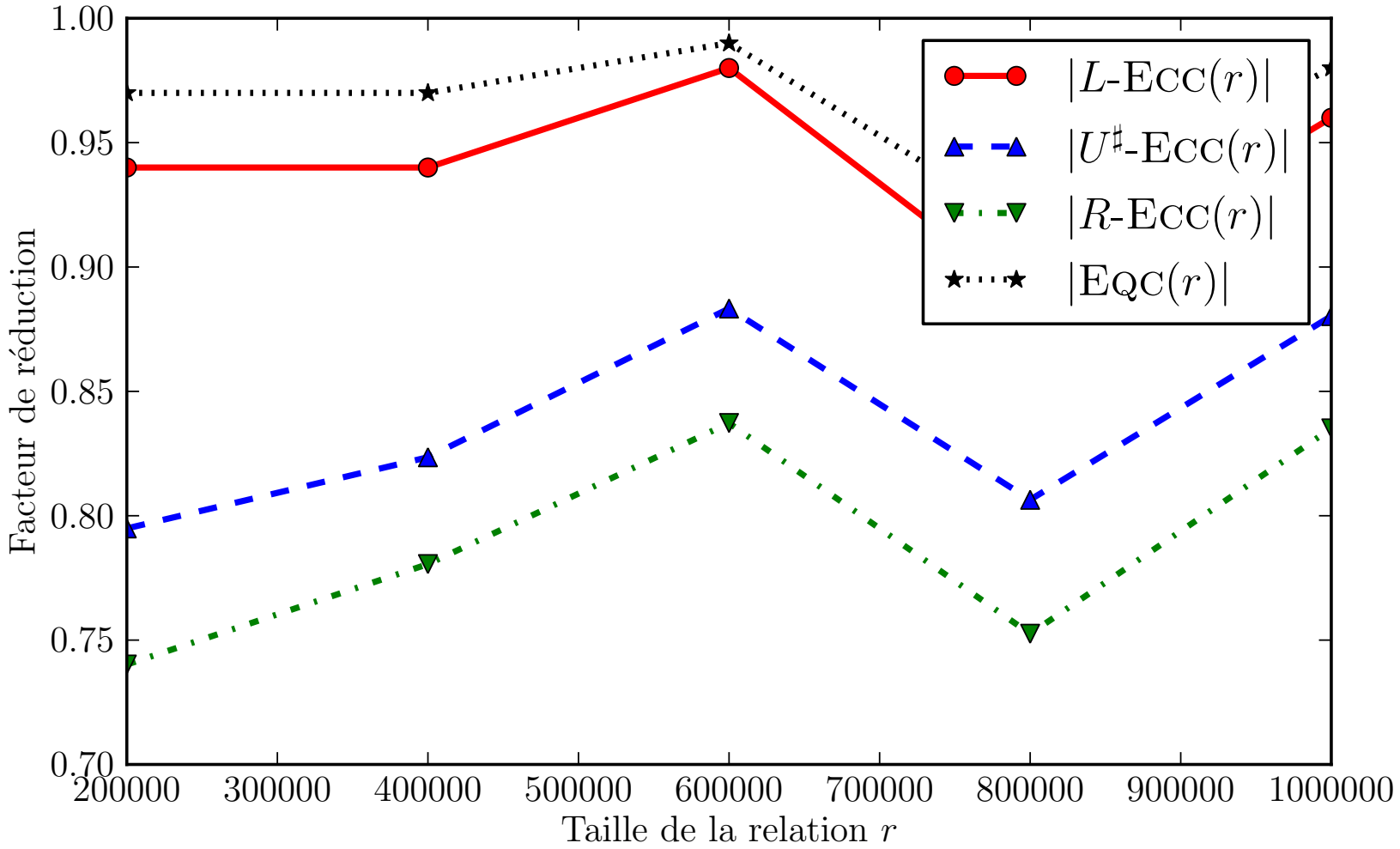
Ratios de réduction avec $\mathcal{D} = 10$, $\mathcal{C} = 100$, $\mathcal{T} = 1000\text{K}$:



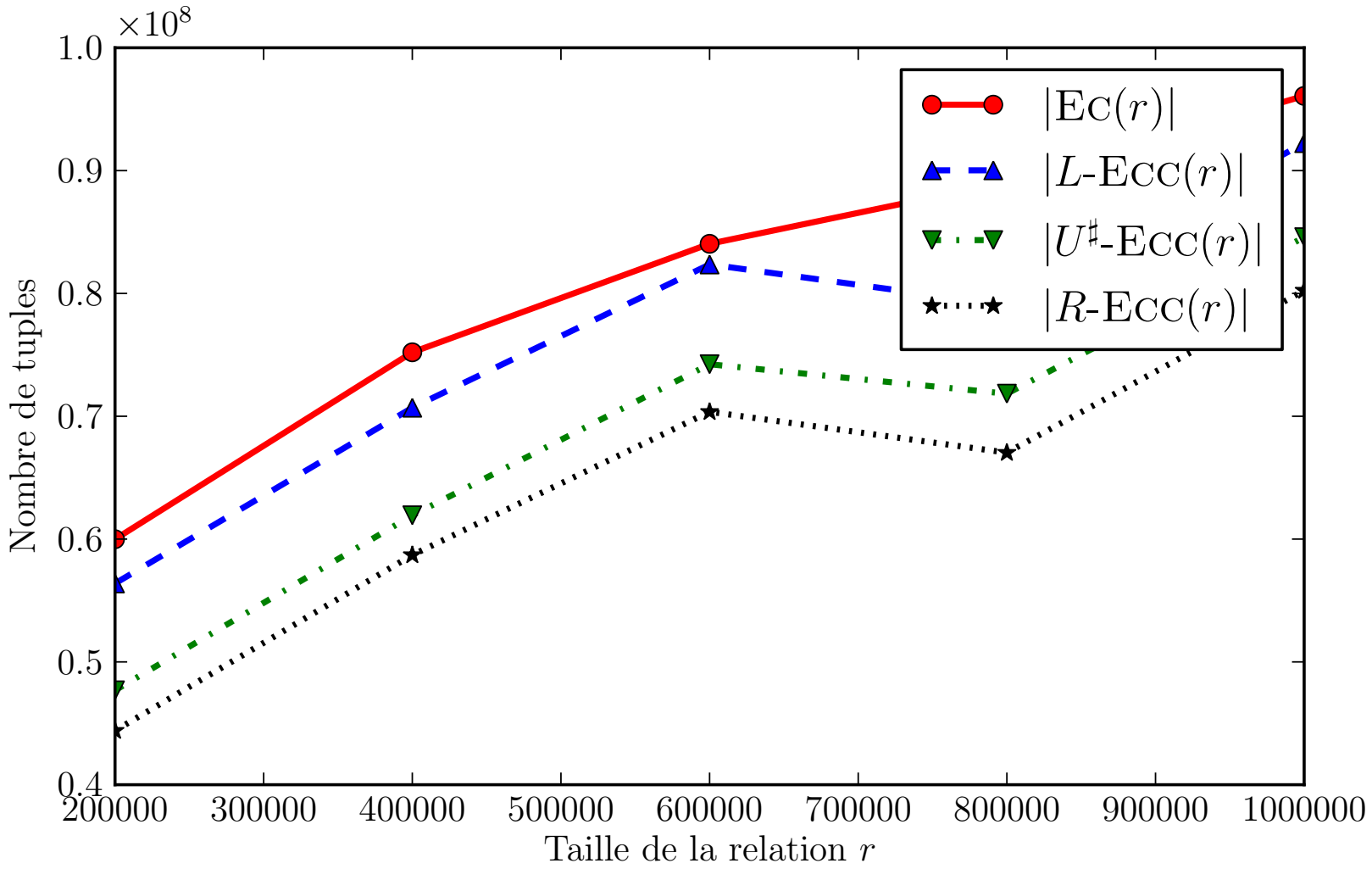
Ratios de réduction avec $\mathcal{D} = 10$, $\mathcal{T} = 1000\text{K}$, $\mathcal{S} = 0$:



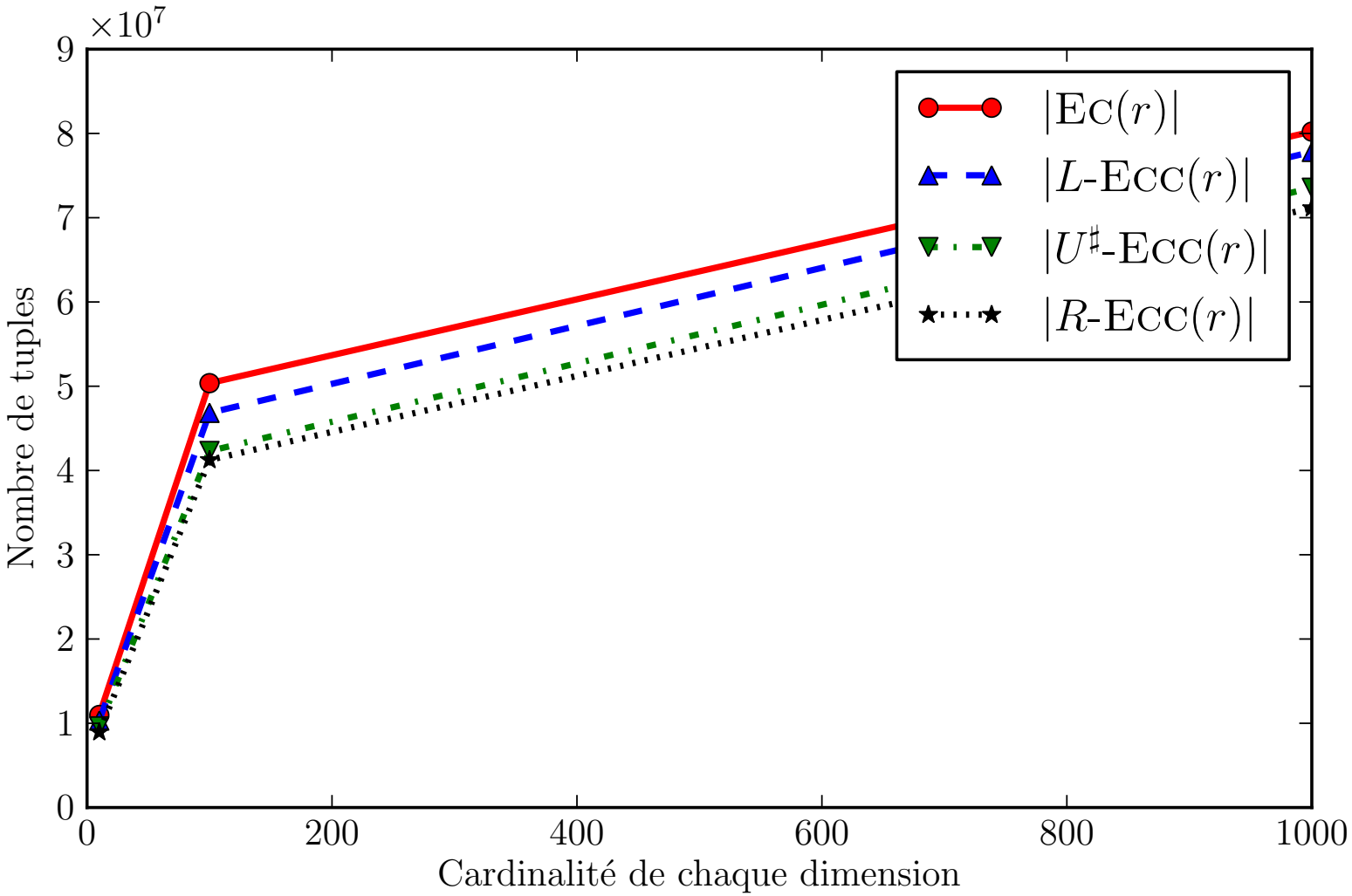
Ratios de réduction avec $\mathcal{D} = 10$, $\mathcal{C} = 100$, $\mathcal{S} = 0$:



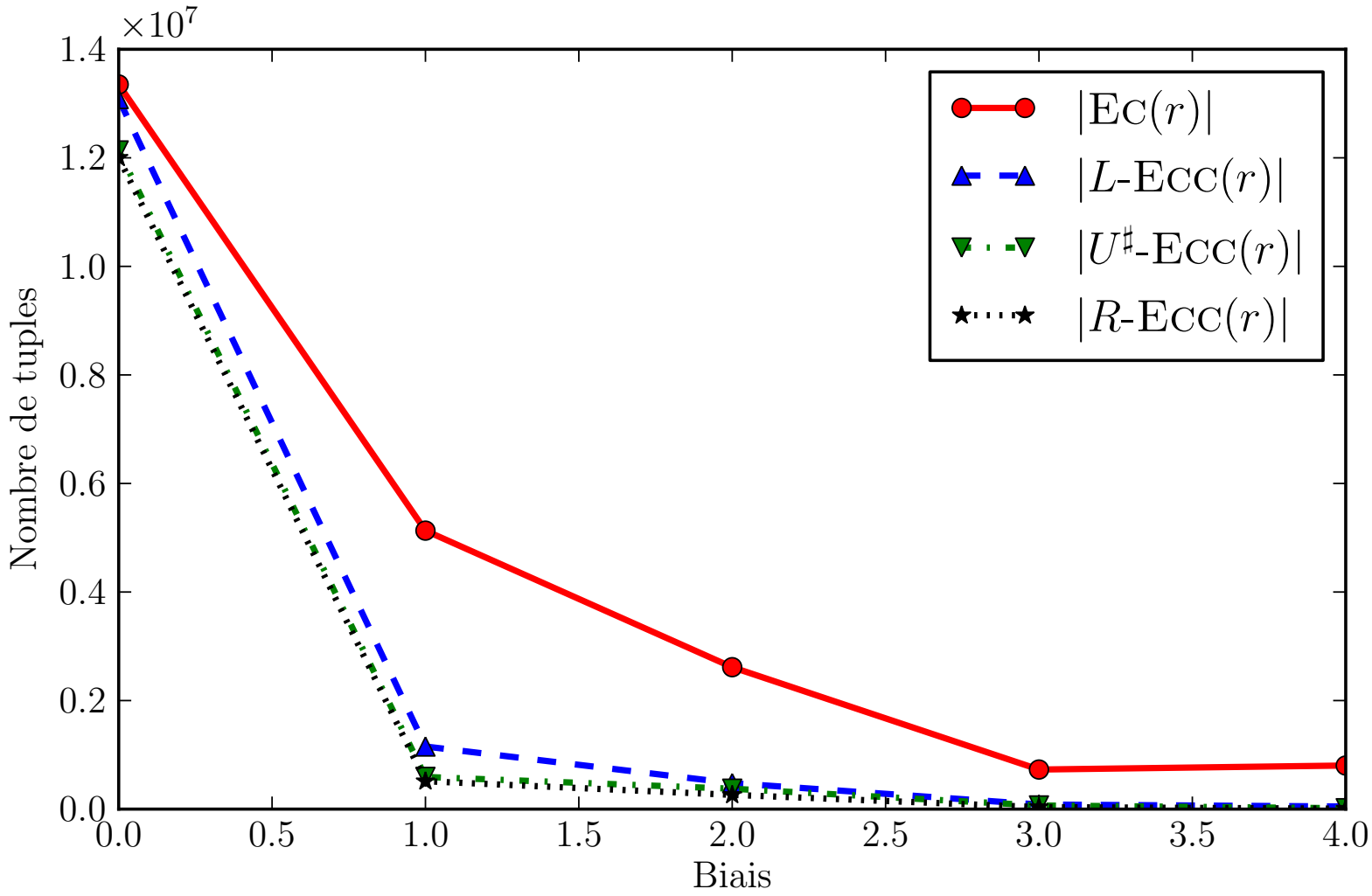
Taille du Cube Fermé Émergent avec $\mathcal{D} = 10$, $\mathcal{C} = 100$, $\mathcal{S} = 0$



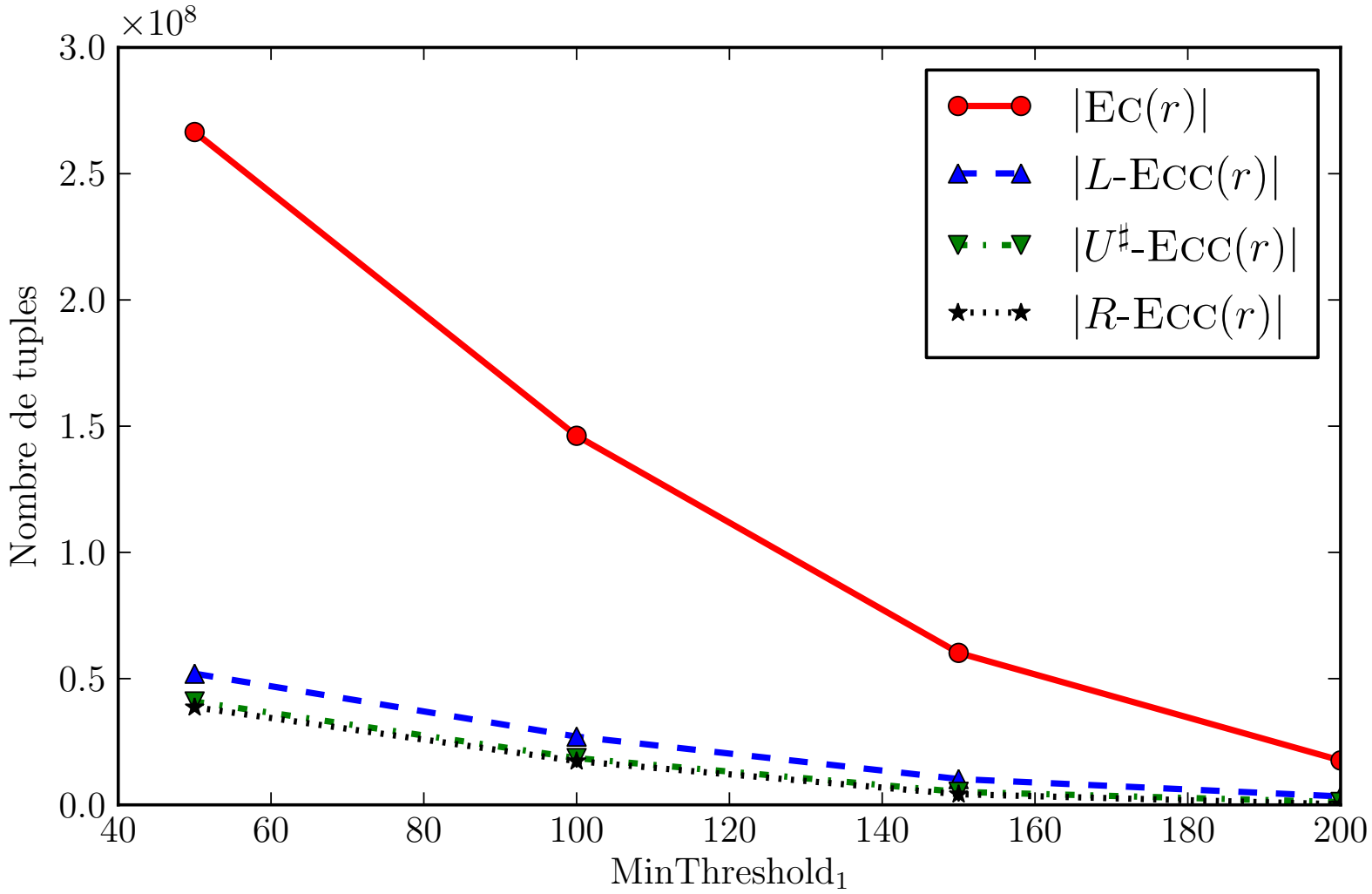
Taille du Cube Fermé Émergent avec $\mathcal{D} = 10$, $\mathcal{T} = 1000$K, $\mathcal{S} = 0$



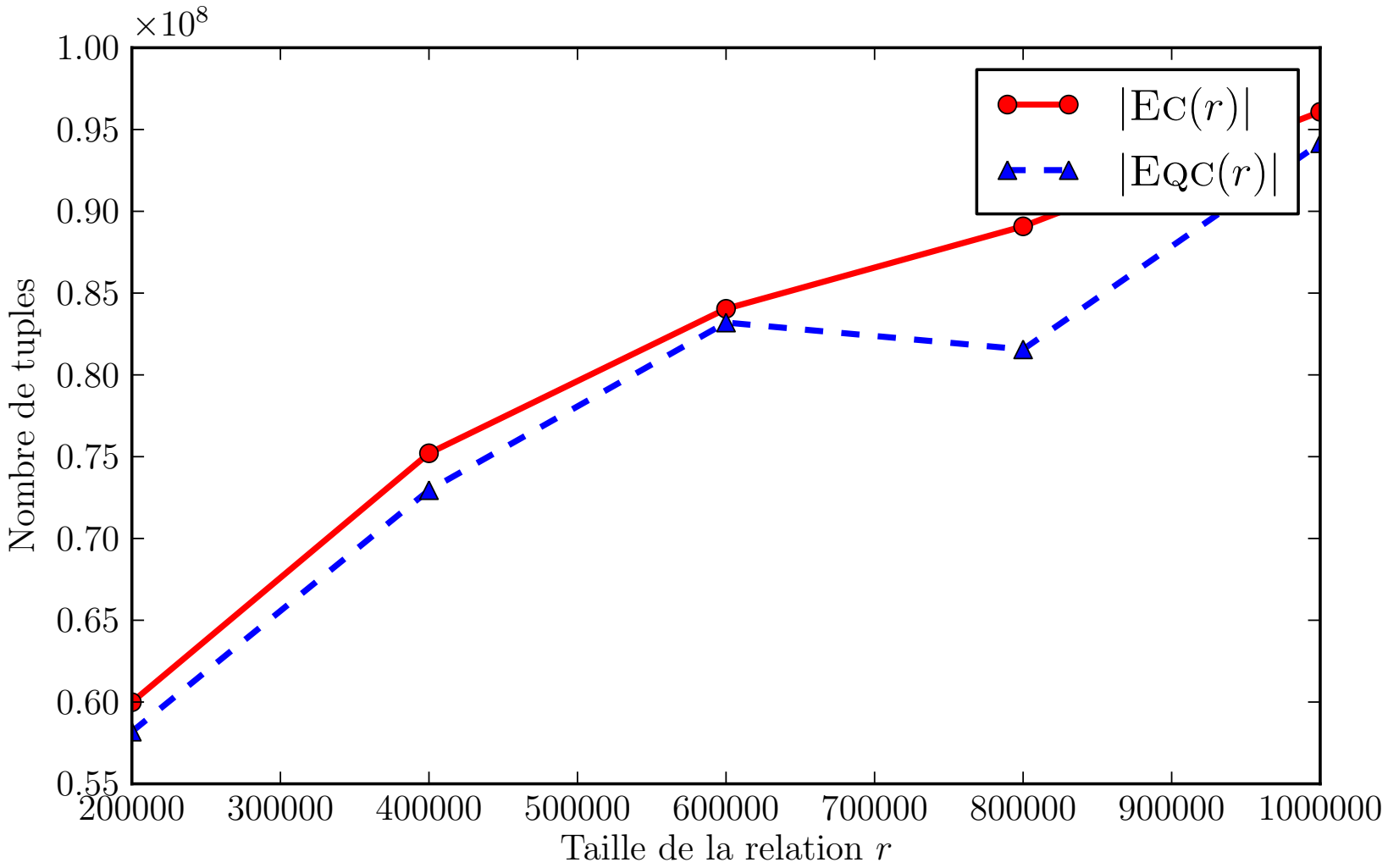
Taille du Cube Fermé Émergent avec $\mathcal{D} = 10$, $\mathcal{C} = 100$, $\mathcal{T} = 1000$K



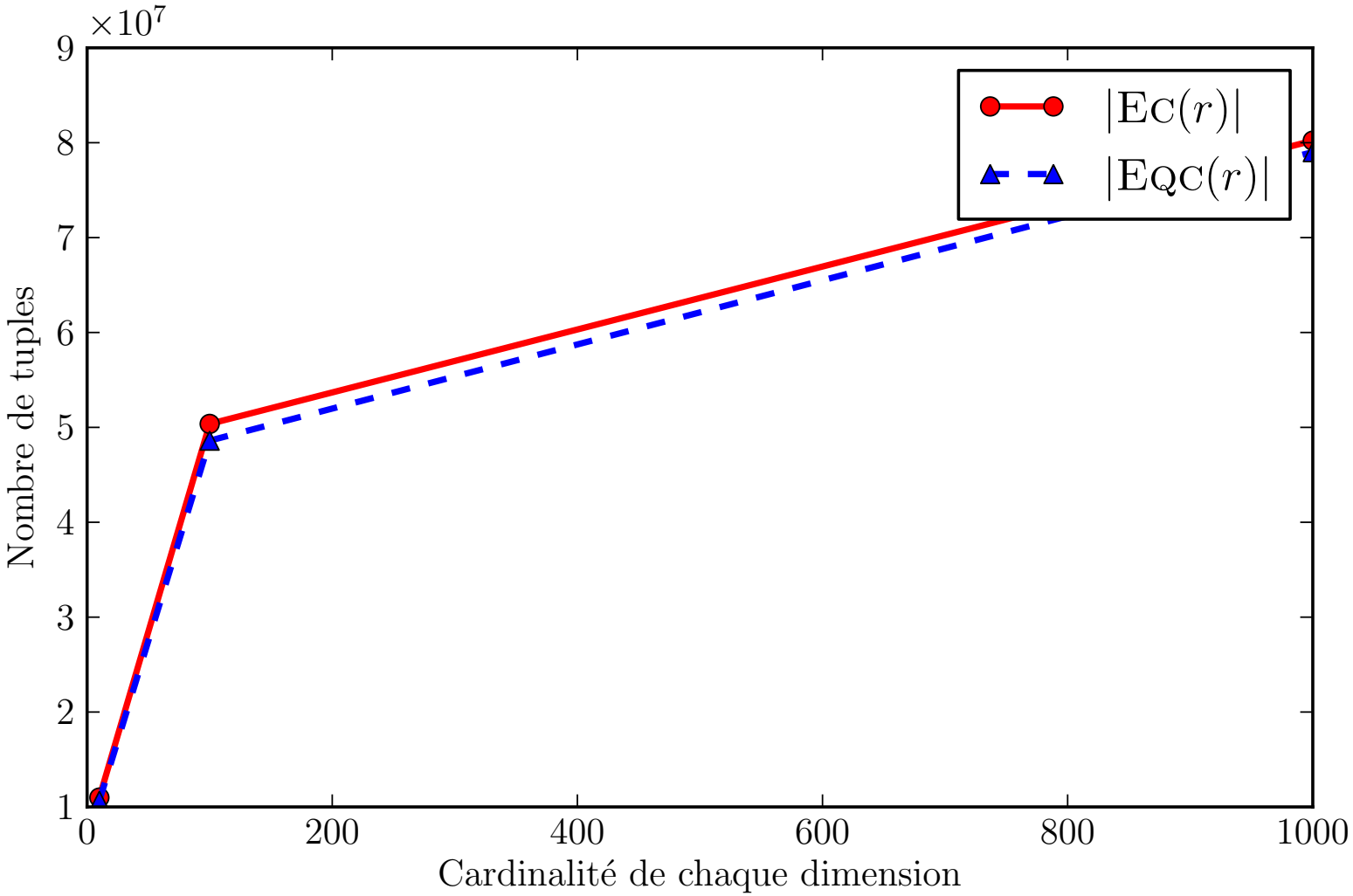
Taille du Cube Fermé Émergent pour les relations de données météorologiques



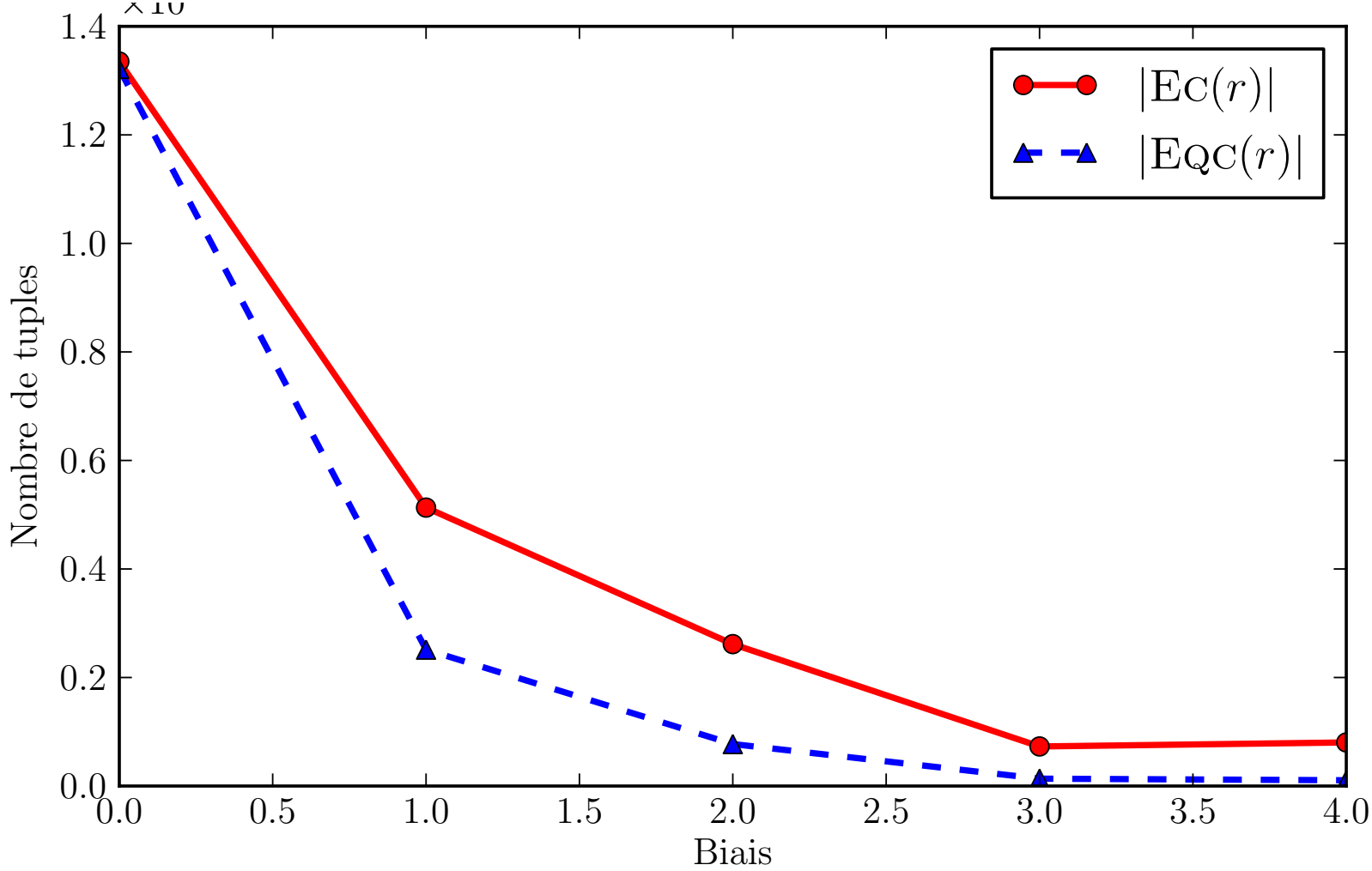
Taille du Cube Quotient Émergent avec $\mathcal{D} = 10$, $\mathcal{C} = 100$, $\mathcal{S} = 0$



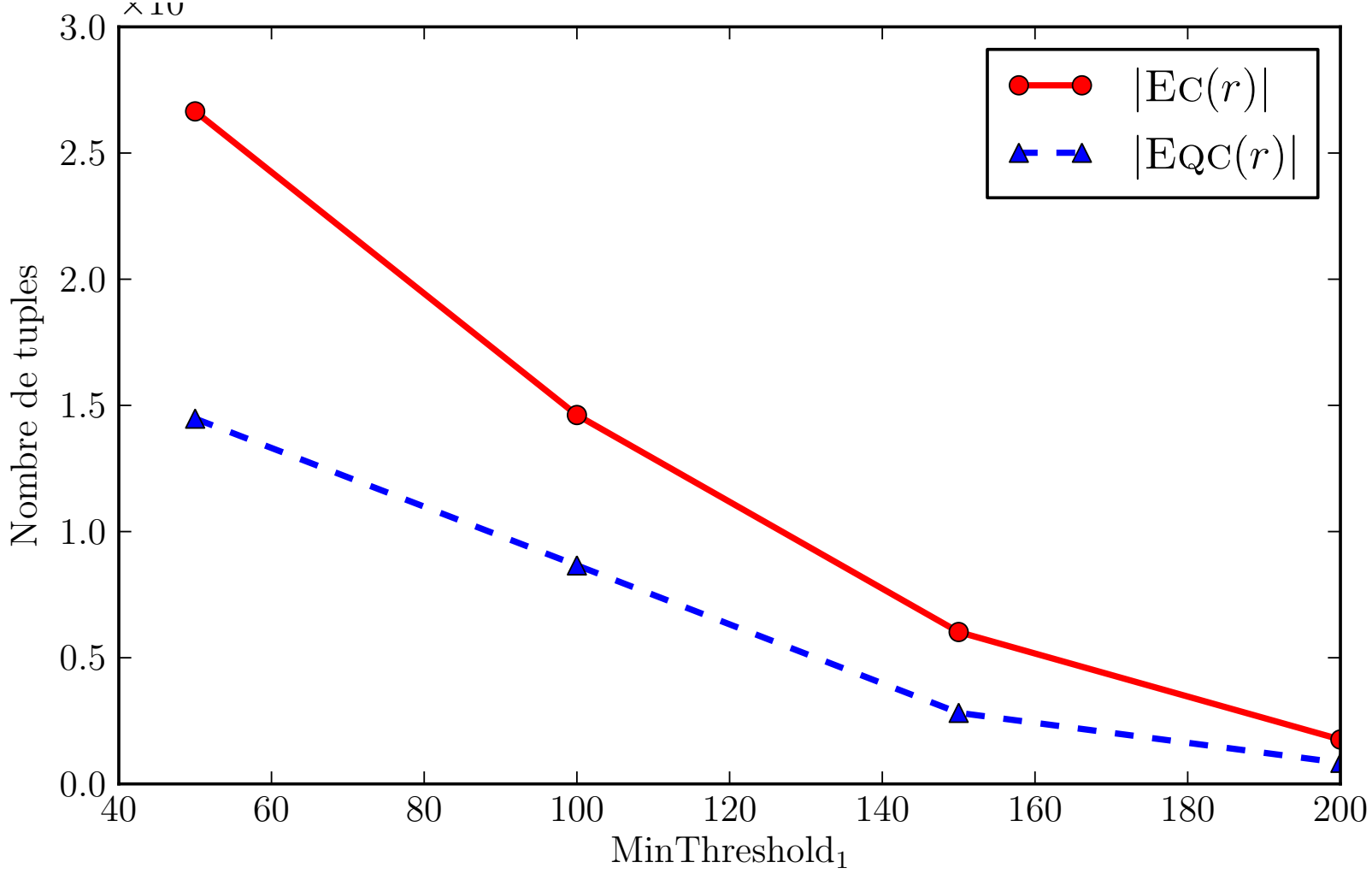
Taille du Cube Quotient Émergent avec $\mathcal{D} = 10$, $\mathcal{T} = 1000\text{K}$, $\mathcal{S} = 0$



Taille du Cube Quotient Émergent avec $\mathcal{D} = 10$, $\mathcal{C} = 100$, $\mathcal{T} = 1000K$



Taille du Cube Quotient Émergent pour les relations de données météorologiques



Idées du comptage probabiliste

- Une fonction de hachage $h : CL(r) \mapsto \{0,1\}^n$ associe à chaque tuple de l'espace de recherche une valeur ayant l'aspect de l'aléa.
- Sur l'ensemble des valeurs hachées on prend une observable M qui ne dépend que de l'ensemble sous-jacent, c'est à dire ni des réplications, ni des permutations. Dans HyperLogLog, M a pour valeur le maximum des positions du premier 1.
- Pour améliorer la précision de l'observable, on simule m expériences indépendantes en divisant les données en m lots.
- L'estimateur est obtenu en faisant la moyenne (harmonique) des m observables et en la normalisant.