



Évaluation T0+24 MARMOTE

MARkovian MOdeling Tools and Environments

<https://wiki.inria.fr/MARMOTE/Accueil>

Contenu

Qui sommes-nous ?

Qu'avons-nous réalisé ?

- Résultats théoriques

- Résultats des applications

- Résultats logiciels

- Publications

Comment avons-nous travaillé ?

Qu'allons-nous faire ?

- Intensifier l'ingénierie logicielle

- Développer les applications

- Communiquer

1

Présentation

Qui sommes-nous ?

Partenaires Institutionnels et leur « partner heads » :

- ▶ INRIA/MAESTRO : Alain Jean-Marie, coordinateur
- ▶ INRIA/DYOGENE (anc. TREC) : Ana Bušić
- ▶ INRIA/MESCAL : Bruno Gaujal
- ▶ Univ. Versailles-St-Quentin/PriSM : Jean-Michel Fourneau, co-coordonateur
- ▶ Telecom SudParis/SAMOVAR : Hind Castel-Taleb
- ▶ Univ. Paris-Est-Créteil/LACL : Nihal Pekergin
- ▶ Univ. Paris 6/LIP6 : Emmanuel Hyon

Spécialistes de la *modélisation Markovienne* et de ses applications en télécoms, systèmes distribués, fiabilité...

Issus de la même école : vocabulaire commun, travaux communs antérieurs...

Que voulons-nous faire dans MARMOTE

Objectifs du projet :

- ▶ Faire avancer les méthodes de résolution pour chaînes de Markov
 - ▶ échantillonnage exact de la distribution stationnaire (« simulation parfaite »)
 - ▶ simulation Monte Carlo parallèle
 - ▶ solutions à forme close et algorithmes de résolution de flots
 - ▶ simulation numérique : calcul de bornes, méthodes de transformées, ...
- ▶ Faire avancer les outils logiciels
- ▶ Tester le tout dans plusieurs cas d'application

2

Qu'avons-nous réalisé ?

Théorie

- ▶ Solutions analytiques à forme close pour des réseaux de files d'attente avec signaux et algorithmes adaptés pour les calculs des distributions stationnaires
- ▶ Calcul de borne sur des distributions (meilleure borne st) et applications à l'analyse d'une file soumis à un flux dont on a mesuré une trace.
- ▶ Simulation Parallele pour l'analyse de la fiabilité modélisé par des Dynamic Fault Trees (pas dans la proposition initiale)
- ▶ Simulation Parallèle pour certains modèles spécifiées par Stochastic Process Algebra par l'approche dite du Parallel Prefix, conditions suffisantes d'application et méthodes de simulation.

Applications

- ▶ Analyse des méthodes de Restart par des résultats à forme produit
- ▶ Simulation pour une cellule (WP5.2)
- ▶ Algorithmes d'échantillonnage de la distribution stationnaire d'une bande électronique (WP5.5)

Logiciel : état de l'art

Au démarrage de MARMOTE, plusieurs logiciels d'analyse Markovienne développés par les membres :

- ▶ Psi, Psi3 (Inria/MESCAL)
- ▶ Xborne (UVSQ/PRiSM)
- ▶ ERS (Inria/MAESTRO)

Plan de travail en parallèle :

- ▶ Développer un logiciel-chapeau qui reprendra toutes ces fonctionnalités
- ▶ Poursuivre le développement indépendant de Psi3 et Xborne

Logiciel : Xborne

Réalisations :

- ▶ Logiciel d'analyse d'une file soumis à un flux mesuré par une trace (bientôt disponible, sous matlab)
- ▶ Logiciel de résolution de réseaux de files d'attente modélisant des restart (en liaison avec Free Univ. Berlin)

Logiciel : Psi3

(if any)

Logiciel : MARMOTE

Réalisations :

- ▶ Définition de l'architecture du logiciel MARMOTE
- ▶ Études de cas autour des *scientific workflow systems*
- ▶ Implémentation des objets de base à partir du logiciel ERS
- ▶ Interfaçage avec le logiciel Xborne (UVSQ/PRiSM)

Logiciel : MARMOTE (suite)

Idée générale :

- ▶ Hiérarchie de modèles Markoviens
On-Off **is a** Homogeneous Birth-Death **is a** General Birth-Death **is a** Continuous-Time Markov Chain
- ▶ Familles de métriques
Distribution : stationnaire, transitoire, de temps d'atteinte, etc.
- ▶ Pour chaque modèle et chaque métrique, des méthodes de résolution (calcul exact, approché, bornes) adaptées.
⇒ les rendre accessibles par une API

Logiciel MARMOTE (fin) : exemple de code

Comparaison de calculs de la distribution stationnaire

```
// specific methods for F81 1
Felsenstein81* c1 = new Felsenstein81(...);
Distribution* d1 = c1->stationaryDistribution();
Distribution* d2 = c1->simulateChain(...)->getDistribution();
// generic methods for MCs
MarkovChain* c2 = static_cast<MarkovChain*>(c1);
Distribution* d3 = c2->stationaryDistribution_GaussSeidel();
Distribution* d4 = c2->stationaryDistribution_PowerMethod();
Distribution* d5 = c2->stationaryDistribution_Xborne_LowerBound();
Distribution* d6 = c2->replicateSamples_Psi3(...);
Distribution* d7 = c2->simulateChain(...)->getDistribution();
Distribution* d8 = c2->simulateChain(...)->getDistribution();
// comparison
cout << "Distance L1(d1,d2) = " << d1->distanceL1(d2) << endl;
...

```

1. Felsenstein 81 est un modèle utilisé en bio-informatique

Publis

4

Comment l'avons-nous
fait ?

Comment

Personnel :

- ▶ Recrutement xxx, yyy
- ▶ Participation de doctorants hors MARMOTE
- ▶ WP 5.2 : Suite au départ des contacts biologie, changement de cible vers la modélisation de réactions chimiques
- ▶ WP 5.5 : Mise en place d'un groupe de travail sur des modèles de « développement durable » en Économie

Comment (suite)

- ▶ réunions plénières
- ▶ plein de réunions de projet
- ▶ ...

4

Qu'allons-nous faire ?

Planning futur : logiciel

Accélérer sur le développement logiciel :

- ▶ Recruter plusieurs personnes
- ▶ Replanifier l'ingénierie : concaténer les deux ingénieurs WP3 INRIA/MAESTRO, mettre en place un GdT avec les ingénieurs INRIA/MESCAL et UVSQ/PRiSM
- ▶ Sortir une version $0.\beta$ dans les premières semaines de 2015.

Planning futur : applications

Développer les applications :

- ▶ Développer l'application WP5.4 (Laser) dans le logiciel MARMOTE
- ▶ Démarrer l'application « réseaux » (WP 5.1) : le post-doc a été recruté au 1/11/14
- ▶ Démarrer xxx
- ▶ Démarrer yyy

Planning futur : communication

Commencer à diffuser le projet :

- ▶ Communiquer dans les conférences nationales : ROADEF 2015, ...
- ▶ Communiquer à l'international : soumettre à VALUETOOLS, QEST, ...