



Les WP3 et WP4 du projet MARMOTE

Contenu

WP3

Concepts

Formats et Langages

Architecture

Programme de travail

WP4

Bornes pour métriques Markoviennes

Processus semi-Markoviens

Algorithmes pour les MDPs

1

WP3

WP3

WP3 : Development of a modeling environment for Markovian systems

Objectifs

Mettre à disposition du scientifique un « environnement de modélisation » qui doit donner accès aux algorithmes mis au point par les spécialistes. Caractéristiques principales : aussi ouvert que possible, orienté-composants (*plugins*), contributif. L'initialiser avec un langage de modélisation, une interface utilisateur minimale, algorithmes de résolution minimaux.

Structure

- ▶ Responsable : J.-M. Fourneau + A. Jean-Marie
- ▶ Participants : J.-M. Vincent, F. Quessette, S. Tan, E. Hyon, F. Delbot
- ▶ Effort : 48 p×m

Features

Plusieurs scénarios d'utilisation :

- ▶ API
- ▶ manipulation indirecte, programmation visuelle, workflow
- ▶ web service
- ▶ enseignement
- ▶ ...

1.1

Concepts

Concepts

Identifier les concepts (catégories, hiérarchies, objets...) de base.

- ▶ Modèles
Stochastic Process, Semi-Markov Process, Markov Chain, Markov Additive Process, Markov Modulated Process, Birth-Death Process, ...
- ▶ Métriques
Transient Distribution, Stationary Distribution, Spectral Density, Function Average, Function Variance, ...
State, Trajectory, Hitting Time, Sojourn Time, ...
- ▶ Résultats/Sorties
Formulas, Tables, Graphs, Statistics, Experimental Plans, ...
- ▶ Workflows

Vocabulaire Markovien (AS182, nov. 2004)

Point de départ = le vocabulaire courant de la théorie des chaînes de Markov (liste non exhaustive) :

Processus Stochastique	Chaîne de Markov	Temps Discret
Temps Continu	État	Espace d'États
Fini	Infini	Récompense
Variable Aléatoire	Transition	Matrice de transition
Vecteur Propre	Spectre	Probabilité
Quantile	Stationnaire	Transitoire
Distribution	Quantile	Fonction Génératrice
Transformée de Laplace	Moyenne	Variance
Moment	Densité Spectrale	Simulation
Trajectoire	Échantillon	Exact
Approché	Borne	Asymptotique
Réversible	Naissance-et-Mort	Quasi-Naissance-et-Mort

1.2

Formats et Langages

Comment décrire le problème et sa résolution

Nécessité de :

- ▶ décrire les modèles
Espaces d'états, sous-espaces (problèmes d'atteinte, probas taboues...)
Markov (temps continu/discret), Semi-Markov, Rewards, Contrôles (MDPs et jeux)
- ▶ décrire les métriques
Mesures de performance, formules logiques du Model Checking, critères d'optimisation, contraintes diverses
- ▶ décrire les traitements
Enchaînement d'algorithmes, plans d'expérience, traitement des résultats

Formats de matrices/graphes

- ▶ Format textuel simple, plein ou creux (MCL, MARCA_format)

```
discrete sparse
16
0 0 0.9091
0 4 0.0909
1 1 0.9091
...
stop
```

- ▶ Format HBF
- ▶ Formats pour graphes (valués ?) : DOT, YED, TGF, GML, GraphML

Langage MarMoL ?

Langages de plus haut niveau

► Formats/langages symboliques (PEPS, RGL)

Cycle :

```
(i, (i+2)%5, rand(1000)) 0<=i<=4
```

Cube 3D :

```
n:=1
```

```
((2*i,2*j,2*k),(2*(i+1),2*j,2*k),0) -n<=i<n -n<=j<=n -n<=k<=n
```

```
((2*i,2*j,2*k),(2*i,2*(j+1),2*k),0) -n<=j<n -n<=i<=n -n<=k<=n
```

```
((2*i,2*j,2*k),(2*i,2*j,2*(k+1)),0) -n<=k<n -n<=i<=n -n<=j<=n
```

Naissance et mort (infini!) :

```
formal lambda;
```

```
formal mu;
```

```
(n,n+1,lambda) 0<=n
```

```
(n,n-1,mu) 1<=n
```

Potentialités d'un langage

Avantages potentiels :

- ▶ compacité
- ▶ familles de modèles
- ▶ classification
- ▶ traitement formel (formules, ...)
- ▶ composition, décomposition
- ▶ génération adaptée de structures de données

Challenges :

- ▶ classification
- ▶ composition, décomposition
- ▶ génération adaptée de structures de données

1.3

Architecture

Fonctionnalités

Objectifs poursuivis pour l'architecture :

- ▶ travail collaboratif
- ▶ contributions hétérogènes en langages de programmation, interface utilisateur, exigence de licence
- ▶ inclusion de code « legacy »
- ▶ portage

Solutions à étudier

- ▶ INRIA DTK
- ▶ OpenAlea (C/C++ et Python)
- ▶ ... ?

L'algorithme comme objet manipulable

Un algorithme est associé à :

- ▶ entrées
- ▶ contraintes d'application
- ▶ résultats

mais aussi

- ▶ inventeur, références bibliographiques
- ▶ complexité théorique, performances avérées
- ▶ ...

Métriques enrichies

Multiplicité de métriques pour les CMs, Markov Reward Processes,

- ▶ Trajectoires
- ▶ State Distributions : transient, stationary
- ▶ Time Distributions : hitting time, sojourn time
- ▶ Process metrics : correlations, spectral density,

Multiplicité de valeurs possibles pour chaque métrique :

- ▶ Exact : formel, numérique
- ▶ Borne : supérieure, inférieure, asymptotique
- ▶ Approximation : avec borne, sans borne (“heuristique”)

1.4

Programme de travail

La promesse

D'après le document scientifique MARMOTE :

A functional version should be delivered at T0+21, with beta versions along the way.

Autre promesse :

We shall foster the use of the software in teaching among the participants of the project. This effort will be extended at the international level, for instance through the EIT ICT Labs, of which INRIA and UVSQ partners are members.

L'intégration du projet sera avérée si le logiciel est utilisé par les développements du WP 5.

Work Force

Deux ingénieurs prévus intégralement sur ce WP

1. T0+3 → T0+21 (18 p×m) : développement du prototype
⇒ **Issam Rahbi** à partir du 15 octobre 2013
(= T0+9.5 → T0+27.5)
2. T0+33 → T0+45 (12 p×m) : maintenance, prise en compte du retour d'expérience, aide à l'implémentation des méthodes issues des autres WPs

Autres contributions : permanents (Sovanna Tan, Franck Quessette, François Delbot, Emmanuel Hyon, Jean-Marc Vincent, Alain Jean-Marie, Jean-Michel Fourneau), post-doctorant "WP1" et ingénieur "WP5.5".

Fiche de poste 1

Fiche de poste : Programming Engineer, duration 18 months
The engineer will be responsible of the design and the realization of the modeling environment, as described in WP 3. The person will be responsible for collecting requirements from different groups of scientists, propose an architecture for the software and implement the first versions.

Knowledge, Skills and Experience : From a software engineering point of view, the main challenge is to produce a reusable modular and versatile code than will be used by all partners but also (in the longer term) by the community. The engineer should be interested in collaborative software environments.

Diplomas : Master's degree or PhD in computer science/engineering school, plus some experience in large software projects.

Fiche de poste 2

Fiche de poste : Programming Engineer, duration 12 months

The engineer will be responsible for :

- ▶ the implementation of algorithms designed in WP 4 into the software environment ;
- ▶ collaborating with the partners working on WP 5 (use of the modeling environment for different applications) ;
- ▶ The general maintenance and development of the modeling environment, according to the feedback provided by users.

Knowledge, Skills and Experience : The person should be at ease with modeling as well as software development. S/He shall have to adapt fast to the modeling and programming environment, and to the different contexts of several partners. Some experience in research-engineering work will be welcome.

Diplomas : Master's degree in computer science/engineering school

Organisation

Préparation de l'atelier logiciel du mercredi 9 octobre 2013

Développement distribué mais organisé

- ▶ Mise en place d'une "gouvernance" ?
- ▶ Mise en place de la plate-forme collaborative
- ▶ Réunions régulières, suivi du projet ?
- ▶ Coding sprints ?
- ▶ ...

Premières étapes

- ▶ Se mettre d'accord sur le vocabulaire
- ▶ Identifier les façons de travailler avec un logiciel de modélisation (workflows)
- ▶ Inventaire logiciel : mettre à jour les Fiches AS182
 - ▶ langages de programmation utilisés,
 - ▶ niveau de "qualité logicielle"
 - ▶ potentiel de décomposition, réutilisation
 - ▶ formats de données, formats de résultats
- ▶ Identifier les besoins et les ressources humaines
- ▶ Identifier la marche à suivre pour obtenir des résultats montrables rapidement
- ▶ Use cases : comment nous serv(ir)ons-nous du logiciel

2

WP4

WP4 : simulation numérique

Le workpackage fourre-tout pour les développements théoriques dans les méthodes Markoviennes génériques (pas directement liés à une application donnée).

Ces développements ont vocation à être implémentés dans le WP3.

Calcul de Bornes

Spécialité de la maison depuis le XXème siècle :

- ▶ comparaison stochastique de chaînes de Markov
- ▶ comparaison de modèles
- ▶ bornes sur les métriques (probas stationnaires, transitoires, temps d'atteinte...)
- ▶ discrétisation avec garantie de borne
- ▶ troncature avec garantie de borne

SMAP : Semi-Markov Accumulation Processes

Processus d'accumulation (“Markov Reward”) qui généralisent de nombreux modèles connus (MAP, DMAP, etc.)

Ingrédients :

- ▶ un processus semi-markovien à espace discret (“environnement”) $\{Z(n); n \in \mathbb{N}\}$, qui génère
- ▶ une suite de dates $\{T_n; n \in \mathbb{N}\}$;
- ▶ une quantité $Q(t)$ qui s'accumule
 - ▶ par incréments stochastiques aux moments des sauts T_n
 - ▶ selon un processus de Lévy entre les sauts.

Avec ces données, on construit des matrices **A** et **B** telles que la transformée de Laplace espace-temps de $Q(t)$ s'obtient en résolvant :

$$\mathbf{K} = \mathbf{A} + \mathbf{BK}$$

Algorithmes pour Markov Decision Processes

Nombreux challenges pour les MDP et les jeux stochastiques, dont :

- ▶ traitement “formel” (sans développement de l'espace d'états) des MDPs pour contourner la malédiction de la dimension. Ordres, monotonie, comparaisons stochastiques...
- ▶ programmation dynamique approchée mais garantie Troncation de l'espace d'état, troncation de l'horizon de programmation, bornes d'erreur

Au travail !



visitez

www.inria.fr/MARMOTE
(prochainement)