Proceedings of the 39th IEEE
Conference on Decision and Control
Sydney, Australia • December, 2000

Decomposition and Parallel Processing Techniques for Two-Time Scale Controlled Markov Chains[1]

J. A. Filar[2] J. Gondzio[3] A. Haurie[4] F. Moresino[5] J.-Ph. Vial[6]

Abstract

This paper deals with a class of ergodic control problems for systems described by Markov chains with strong and weak interactions. These systems are composed of a set of m subchains that are weakly coupled. Using results recently established by Abbad et al. one formulates a limit control problem the solution of which can be obtained via an associated nondifferentiable convex programming (NDCP) problem. The technique used to solve the NDCP problem is the Analytic Center Cutting Plane Method (ACCPM) which implements a dialogue between, on one hand, a master program computing the analytical center of a localization set containing the solution and, on the other hand, an oracle proposing cutting planes that reduce the size of the localization set at each main iteration. The interesting aspect of this implementation comes from two characteristics: (i) the oracle proposes cutting planes by solving reduced sized Markov Decision Problems (MDP) via a linear programm (LP) or a policy iteration method; (ii) several cutting planes can be proposed simultaneously through a parallel implementation on m processors. The paper concentrates on these two aspects and shows, on a large scale MDP obtained from the numerical approximation "à la Kushner-Dupuis" of a singularly perturbed hybrid stochastic control problem, the important computational speed-up obtained.

1 Introduction

Markov Decision Processes (MDPs) or their control counterpart, *Controlled Markov Chains* (CMCs) are versatile modeling tools benefitting from a rather complete theoretical framework and a series of efficient computational tools. We refer the reader to the books by M. Puterman [21] or D. Bertsekas [4] for a comprehensive presentation of these methods. In the stochastic optimal control realm CMCs play an important role in the numerical solution of problems involving controlled diffusion and jump Markov processes. The book by Kushner and Dupuis [17] gives a comprehensive presentation of these numerical techniques that use approximating CMCs. It has been shown very early (see [7], [18]) that linear programming could be used to solve problems involving MDP's with finite state and action spaces, in particular for the ergodic (average cost) case. As recalled by Blondel and Tsitsiklis in a recent survey of computational complexity results in control [5] ... *linear programming is the only method known to solve average cost MDPs in polynomial time.* However the linear programs (LPs) associated with MDPs are quite large and may suffer from ill-conditioning when the Markov chains contain strong and weak interactions, that is transitions probabilities differing of an order of magnitude and corresponding to different time scales. These problems, related to the theory of singularly perturbed control systems, have been studied by Delebeque and Quadrat [8] and Phillips and Kokotovic[20] among others and, more recently, by Abbad, Bielecki and Filar [1] and [2] who have shown that, in the case of average cost MDPs, a limit control problem could be defined with an associated LP having a nice block-diagonal structure.

One way to deal with large scale but structured LPs consists of implementing a decomposition technique. The most celebrated one being the Dantzig-Wolfe [6] method where an auxiliary nondifferentiable convex programming problem is solved by the Kelley cutting plane method. Recent advances in this domain have permitted the development of a pseudo polynomial time decomposition technique called the *Analytic Center Cutting Plane Method* (ACCPM) [13, 15]. In this paper we exploit the structured LP formulation proposed in [1] and ACCPM to provide a pseudo polynomial time algorithm for solving ergodic MDPs with strong and weak interactions.

2 Ergodic MDP with strong and weak interactions

Consider an MDP with finite state and control sets S and U respectively. A state $s \in S$ is represented by a pair $s = (x, i)$ where $x \in X$ corresponds to *a fast mode*

[1] Research partly supported by the Swiss NSF grants # 12-42301.94 - # 12-57093.99/2 and by ARC (Australia) grant # A49906732.

[2] Professor, School of Mathematics, Uni. of South Australia.

[3] Professor, Department of Mathematics & Statistics, The University of Edinburgh, UK.

[4] Professor, HEC, University of Geneva, Switzerland.

[5] Research Associate, The Judge Institute of Management Studies, University of Cambridge, Cambridge, UK.

[6] Professor, HEC, University of Geneva, Switzerland.

0-7803-6638-7/00$10.00 © 2000 IEEE

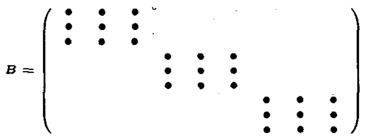
and $i \in I$ to a *slow mode* respectively. More precisely, the generator of the MDP $G^{\varepsilon}[(x,i),(x',j)|u]$ is assumed to have the form

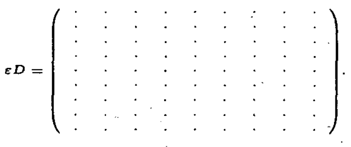
$$\begin{aligned} G^{\varepsilon}[(x,i),(x',j)|u] = B[(x,i),(x',j)|u] + \\ \varepsilon D[(x,i),(x',j)|u] + o(\varepsilon) \end{aligned} \tag{1}$$

where

- $B[(x,i),(x',j)|u]$ is the generator of a completely decomposable MDP, with card(I) subprocesses which do not communicate one with the other; *i.e.* if $i \neq j$ then $B[(x,i),(x',j)|u] \equiv 0 \quad \forall x, x' \in X$
- $\varepsilon D[(x,i),(x',j)|u]$ is a perturbation that links together these card(I) sub-blocks.

This structure is illustrated below by the shadows of the matrices B and εD respectively. The size of the dot indicates the order of magnitude of the corresponding coefficient





A transition cost $L^i(x,u)$ is associated with the state $s = (i,x)$ and the control action u. The class Γ of admissible stationary policies is a set of feedback laws $\gamma : (x,i) \mapsto u = \gamma(x,i)$. We assume, to simplify, that each admissible policy generates an ergodic Markov chain. One searches for a stationary policy $\gamma^* \in \Gamma$ that maximizes the average cost criterion

$$J(\gamma) = \liminf_{N\to\infty} \frac{1}{N} E_{\gamma}[\sum_{n=0}^{N-1} L^{i(n)}(x(n),u(n))]. \tag{2}$$

2.1 LP formulations

Let $Z^i(x,u)$ denote the joint probability of being in state (i,x) and taking the action u. The MDP can be solved as the following LP

$$\begin{aligned} \max \quad & \sum_i \sum_x \sum_u L^i(x,u) Z^i(x,u) && (3) \\ \text{s.t.} \\ 0 \quad = \quad & \sum_{i\in I} \sum_{x\in X} \sum_{u\in U} G^{\varepsilon}[(x,i),(x',j)|u] Z^i(x,u) \\ & x' \in X, \quad j \in I && (4) \\ 1 \quad = \quad & \sum_i \sum_x \sum_u Z^i(x,u) && (5) \\ 0 \quad \leq \quad & Z^i(x,u) \quad i \in I \quad x' \in X, \quad u \in U. && (6) \end{aligned}$$

This problem is usually ill-conditioned because of the presence of big and small transition probabilities in Eq. (4).

2.2 The LP associated with the limit control problem

It has been shown in [2] that, when ε tends to zero, the solution of the MDP can be approximated by the solution of a *limit control problem* (LCP). To this LCP corresponds the following LP

$$\begin{aligned} \max \quad & \sum_i \sum_x \sum_u L^i(x,u) Z^i(x,u) && (7) \\ \text{s.t.} \\ 0 \quad = \quad & \sum_x \sum_u B[(x,i),(x',i)|u] Z^i(x,u) \\ & \forall x' \in X, i \in I && (8) \\ 0 \quad = \quad & \sum_i \sum_{x'} \sum_x \sum_u D[(x,i),(x',j)|u] Z^i(x,u) \\ & \forall j \in I && (9) \\ 1 \quad = \quad & \sum_i \sum_x \sum_u Z^i(x,u) && (10) \\ 0 \quad \leq \quad & Z^i(x,u) \quad i \in I \quad x' \in X, \quad u \in U. && (11) \end{aligned}$$

In this formulation the ε term has vanished and therefore the ill-conditioning, due to the mixing of high and low transition probabilities, also disappears. In this LP the constraints (8) have a *block-diagonal structure* while the constraints (9-10) are the *coupling* ones. Clearly this places the problem in the realm of decomposition methods in LP.

3 The decomposition method

3.1 Decoupling the MDPs

The LP associated with the limit MDP (7-11) admits a dual formulation. Let $\phi(x',i)$ and $\psi(j)$ be the dual variables associated with the constraint of index (x',i) in Eq. (8) and the constraint of index j in Eq. (9) respectively. Let Υ be the dual variable associated with the constraint Eq. (10). The dual LP is

$$\begin{aligned} & \min_{\psi,\phi,\Upsilon} \Upsilon \\ & \text{s.t.} \\ & \Upsilon \geq L^i(x,u) - \sum_{x'} B[(x,i),(x',i)|u]\phi(x',i) \\ & \qquad - \sum_j \sum_{x'} D[(x,i),(x',j)|u]\psi(j) \qquad (12) \\ & \qquad i \in I,\, x \in X, u \in U. \end{aligned}$$

Define the expressions

$$\begin{aligned} \Pi(\psi,x,i,u) \quad = \quad & L^i(x,u) - \sum_j \sum_{x'} D[(x,i),(x',j)|u]\psi(j), \\ & i \in I,\, x \in X, u \in U. \qquad (13) \end{aligned}$$

Then the the dual problem (12) can be rewritten as

$$\begin{array}{rcll} & & \min\limits_{\psi,\phi,\Upsilon} \Upsilon & \\ & & \text{s.t.} & \\ \Upsilon & \geq & \Pi(\psi,x,i,u) - \sum\limits_{x'} B[(x,i),(x',i)|u]\phi(x',i) & \\ & & i \in I,\, x \in X, u \in U. & (14) \end{array}$$

Due to the block-angular structure of the generator B, the constraints in (14) decouple. More precisely we formulate the card(I) subproblems

$$\begin{array}{rcll} \chi_i(\psi) & = & \min\limits_{\phi(\cdot,i),\Upsilon^i} \Upsilon^i & \\ & & \text{s.t.} & \\ \Upsilon^i & \geq & \Pi(\psi,x,i,u) - \sum\limits_{x'} B^i[x,x'|u]\phi(x',i), & \\ & & \forall x \in X, \forall u \in U, & (15) \end{array}$$

where B^i denotes the block of nonzero coefficients in the matrix B. For each $i \in I$, the problem (15) corresponds to a decoupled MDP. Hence, if one knows the correct dual values $\psi(i)$, $i \in I$, then, by introducing the modified rewards $\Pi(\psi,x,i,u)$ as defined in (13), the problem can be decomposed into card(I) decoupled MDPs. The solution is therefore obtained by solving the convex optimization problem involving the dual variables ψ

$$\min_{\psi \in \mathbf{R}^{\text{card}(I)}} \chi(\psi) \qquad (16)$$

where $\chi(\psi)$ is the convex function defined as $\chi(\psi) = \max_{i \in I} \chi_i(\psi)$ and $\chi_i(\psi)$ is the optimal value obtained for the problem (15). We notice here that, each problem (15) can be solved either as an LP or via a typical dynamic programming method like, e.g. *policy improvement* or *value iterations.*

4 The Analytic Center Cutting Plane Method

We use ACCPM with a parallel processor implementation to solve the convex programming problem (16). The epigraph of χ can be approximated by intersections of half-spaces. Given a test value $\tilde{\psi}$ in $\mathbf{R}^{\text{card}(I)}$, a procedure called *oracle* generates a subgradient $X(\tilde{\psi}) \in \partial\chi$ at $\tilde{\psi}$ with the property

$$\chi(\psi) \geq \chi(\tilde{\psi}) + \langle X(\tilde{\psi}), \psi - \tilde{\psi} \rangle. \qquad (17)$$

This inequality defines a supporting hyperplane for the function to be optimized; we call it an *optimality cut.*

Suppose the oracle has been called at a given sequence of points $\{\psi^n\}$, $n \in N$. The oracle has therefore generated a set of optimality cuts defining a piecewise linear approximation $\underline{\chi} : \mathbf{R}^{\text{card}(I)} \to \mathbf{R}$ to the convex function χ

$$\underline{\chi}(\psi) = \max_{n \in N}\{\chi(\psi^n) + \langle X(\psi^n), \psi - \psi^n \rangle\}. \qquad (18)$$

This permits us to write the following linear program

$$\begin{array}{ll} \min & \varsigma \\ \text{s. t.} & \varsigma \geq \chi(\psi^n) + \langle X(\psi^n), \psi - \psi^n \rangle, \quad \forall n \in N, \end{array}$$

the solution of which gives a lower bound π_l for the convex problem (16). Observe also that the best feasible solution in the generated sequence provides an upper bound π_u for the convex problem (16), *i.e.*

$$\pi_u = \min_{n \in N}\{\chi(\psi^n)\}. \qquad (19)$$

For a given upper bound π, we call *localization set* the following polyhedral approximation

$$\begin{array}{rcll} \mathcal{L}(\pi) & = & \{(\varsigma,\psi) : \quad \pi \geq \varsigma, \quad \varsigma \geq \chi(\psi^n) + & \\ & & \langle X(\psi^n), \psi - \psi^n \rangle, \quad \forall n \in N\}. & (20) \end{array}$$

It is the best (outer) approximation of the optimal set in (16) in the epigraph space of the function χ.

We can now summarize the ACCPM algorithm
1. Compute the analytical center[1] $(\bar{\varsigma}, \bar{\psi})$ of the localization set $\mathcal{L}(\pi_u)$ and an associated lower bound $\underline{\pi}$.
2. Call the oracle at $(\bar{\varsigma}, \bar{\psi})$. The oracle returns one or several cuts and an upper bound $\chi(\bar{\psi})$
3. Update the bounds: $\pi_u = \min\{\chi(\bar{\psi}), \pi_u\}$ and $\pi_l = \max\{\underline{\pi}, \pi_l\}$
4. Update the upper bound π in the definition of the localization set (20) and add the new cuts.

These steps are repeated until a point is found such that $\pi_u - \pi_l$ falls below a prescribed optimality tolerance. The polynomial convergence of the method was studied in [3] and [12].

In our case, as the function χ is the maximum of card(I) functions, *i.e.*

$$\chi(\psi) = \max_{i \in I} \chi_i(\psi), \qquad (21)$$

the oracle may generate multiple cuts, one for each i in I. The single cut (17) is replaced with the following card(I) cuts:

$$\chi(\psi) \geq \chi_i(\tilde{\psi}) + \langle X_i(\tilde{\psi}), \psi - \tilde{\psi} \rangle, \qquad (22)$$

where $X_i(\tilde{\psi})$ is a subgradient of the function χ_i at $\tilde{\psi}$. This multiple cut approach is more efficient than a single cut approach since the computation time to introduce a cut is negligible and the work of the oracle is the same in both cases. Indeed the oracle has to compute, at each iteration, $\chi_i(\psi)$ and $X_i(\tilde{\psi})$ for all subproblems i in I. In the single cut approach one selects the cut touching the epigraph of χ and one doesn't use the other cuts, contrarily to the multiple cut approach where all

[1] The analytic center is the unique point in that set that maximizes the product of the slacks to the defining constraints.

cuts are used. Furthermore, the oracle can benefit from a parallel implementation, since the card(I) MDPs are totally decoupled and, therefore, can be solved on different computers. As indicated earlier, the oracle can use a policy improvement (PI) algorithm instead of a pure LP approach, as it has been observed in practice that PI is efficient in solving average cost MDPs of moderate size.

5 Experimentation

5.1 A switching diffusion ergodic control model

We consider an ergodic stochastic control problem of the class studied in Ref. [11]. More details on the economic interpretation of this model can be found in [19] and [16]. The system has an hybrid state (x, ξ). The continuous state x takes value in $\mathbf{R}^2$ and evolves according to a controlled diffusion process

$$dx_k(t) = [u_k(t) - \alpha_k x_k(t)]dt + \sigma_k d\omega_k(t) \quad k = 1, 2.$$

whereas the discrete state ξ takes value in a finite set I and evolves according to a controlled jump process with transition rates

$$\begin{aligned} \varepsilon q_{i(i+1)}(x_1, x_2) &= \varepsilon\left(E_i - e_i Y(x_1, x_2)\right) \\ \varepsilon q_{i(i-1)}(x_1, x_2) &= \varepsilon\left(D_i + d_i Y(x_1, x_2)\right). \\ &\text{with} \\ Y(x_1, x_2) &= \left(\eta[x_1]^{-\beta} + (1-\eta)[x_2]^{-\beta}\right)^{-\frac{1}{\beta}}, \end{aligned}$$

A reward rate is defined by

$$\begin{aligned} &L^{\xi(t)}(x_1(t), x_2(t), u_1(t), u_2(t)) \\ &\quad = c(\xi(t))Y(x_1(t), x_2(t)) \\ &\qquad -a_1x_1(t) - a_2x_2(t) - A_1x_1^2(t) - A_2x_2^2(t) \\ &\qquad -b_1u_1(t) - b_2u_2(t) - B_1u_1^2(t) - B_2u_2^2(t), \end{aligned}$$

We consider the above model with a set of parameter values given in Table 1. Here the parameter h defines the grid mesh for the x variables, and h_u the grid mesh *for the controls.*

	$c(1) = 1.3$		$\alpha_1 = \alpha_2 = 0.05$
$\nu = 1.0$	$c(2) = 1.6$		$\sigma_1 = \sigma_2 = 3.0$
$\eta = 0.5$	$c(3) = 1.9$		$x_1^{\max} = x_2^{\max} = 100$
$\beta = -0.6$	$c(4) = 2.2$		$x_1^{\min} = x_2^{\min} = 0$
$a_1 = a_2 = 0.4$	$e_i = 0.002$	$\forall i \in I$	$h = 10/3$
$A_1 = A_2 = 0.004$	$E_i = 0.4$	$\forall i \in I$	$u_1^{\max} = u_2^{\max} = 10$
$b_1 = b_2 = 0$	$d_i = 0.004$	$\forall i \in I$	$u_1^{\min} = u_2^{\min} = 0$
$B_1 = B_2 = 0.05$	$D_i = 0.15$	$\forall i \in I$	$h_u = 2$

Table 1: List of parameter values for the numerical experiments.

In this model the parameter ε will eventually tend to zero, leading to a singularly perturbed switching diffusion control problem of the type discussed in [10].

5.2 The approximating MDP

To compute numerically the solution to this stochastic control problem we implement the method of [17] which uses a sequence of approximating MDPs. The singular perturbation structure in the stochastic control problem translates into an MDP with strong and weak interactions, in this approximation technique. We refer again to [10, 16, 19] for more details on the structure of the approximating MDPs.

5.3 Solving the limit control MDP

We implemented two methods for the resolution of the limit control problem: the decomposition method presented above and a direct solution of the structured LP using a commercial solver (CPLEX).

We implemented ACCPM with a policy improvement (PI) algorithm for the oracle[2]. The parallel implementation of ACCPM has been realized using MPI, a library of C-callable routine (see MPI's reference book [22]) on a cluster of 4 PC.

For $I = \{1, 2, 3, 4\}$ and a grid mesh h=10/3, corresponding to 30 sampling points on each axis, we have solved the limit control problem obtained when $\varepsilon \to 0$. The corresponding linear program has 3'849 rows 135'444 columns and 1'136'514 non-zero elements. To solve the LP, CPLEX used 981 seconds (on a PC, 400 Mhz, under Linux)[3]. When using ACCPM-PI the computing time to solve the limit control problem falls to 290 seconds. If we run the parallel version of ACCPM-PI on four processors the execution time drops to 91 seconds. In Figure 1 we display, for the parallel implementation of the decomposition method, the speed-up as a function of the number of processors.

Figure 2 shows, for both methods (CPLEX vs ACCPM-PI), the steady state probabilities for x, when $i = 3$. We see distinctly that both methods give the same results. In addition, the maximal expected reward growth rate J equals 27.6, for both methods. Although the linear programming direct approach gives an accurate solution concerning the steady state probabilities and the maximal expected reward growth rate, this method gives, in most cases, an imprecise solution concerning the controls and the value function. Figure 3 shows the value function and Figure 4 shows the optimal policy, for the discrete state $i = 3$, for both methods. We see that the direct approach gives an accurate result in the middle of the grid but a blurred result near the

[2] The oracle is written in C and uses a sparse linear equation solver SuperLU [9].

[3] CPLEX offers three methods, namely the simplex, the dual simplex and an interior point method. The solver took 981 seconds with the dual simplex, more than 3'000 seconds with the primal simplex. The interior point method of CPLEX stopped after 307 seconds and proposed an infeasible solution with an objective value close to the optimal value. Running the crossover to obtain a feasible solution took 726 seconds more.

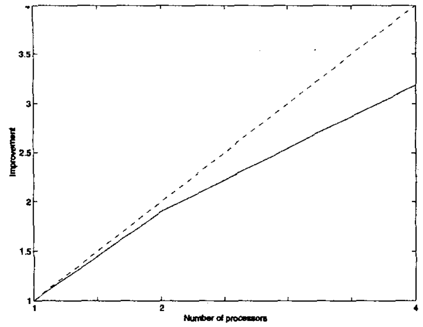
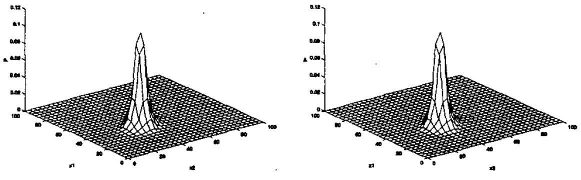


Figure 1: Speed-up as a function of the number of processors.



ACCPM-PI CPLEX

Figure 2: Steady state probability.

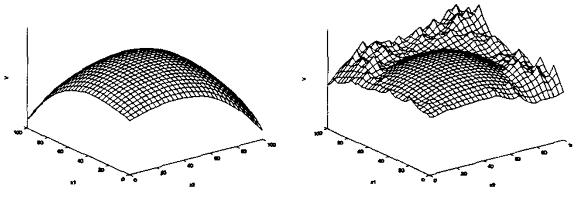
boundaries. This is typically due to the fact that the steady state probabilities are close to 0 near the boundaries. In an LP approach the policy is defined by the ratio of the joint state-action probability $Z^i(x, u)$ with the steady state probability $\sum_{u \in U} Z^i(x, u)$. This ratio is prone to numerical instability when the values are close to 0. The PI algorithm avoids such a pitfall.

5.4 Computational performance

We have performed a series of tests on different instances of this approximating MDP, as shown in Table 2. There the value in first column, e.g. D10-2, indicates the number of sampling points on each x-axis (in this case 10) and the number of discrete states (here card(I)=2). The other columns indicate the size of the associated LP for the limit control problem.

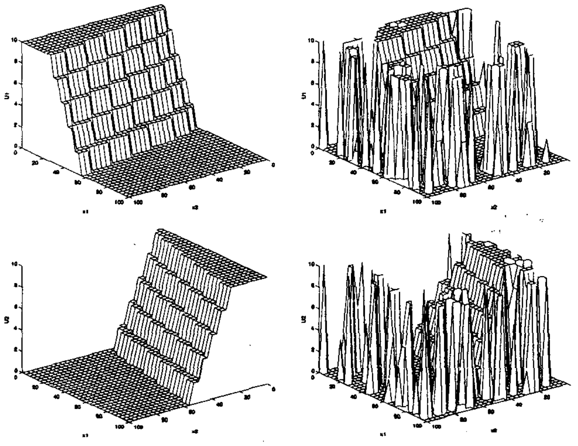
Problem	# subproblems	rows	cols	nonz
D10-2	2	245	8202	63036
D30-2	2	1925	67722	534396
D50-2	2	5205	184842	1466556
D70-2	2	10085	359562	2859516
D10-4	4	489	16404	134274
D30-4	4	3849	135444	1136514
D50-4	4	10409	369684	3117954
D10-6	6	733	24606	205512
D30-6	6	5773	203166	1738632
D40-6	6	10093	357246	3066792
D10-12	12	1465	49212	419226
D30-12	12	11545	406332	3544986

Table 2: Characteristics of the equivalent LP formulation. (size reduced by the presolver).



ACCPM-PI CPLEX

Figure 3: Value function.



ACCPM-PI CPLEX

Figure 4: Optimal policy.

The corresponding execution times are given in Table 3, whereas the speed-up resulting from prallel implementation are reported, for the different models, in Table 4. Finally Table 5 indicates,for different instances of the problem, the maximal size of the grid that would lead to a computationally feasible limit control model. For instance a problem with 4 discrete states and 200 sampling points on each x-axis was solved in 18h36 using four processors. The maximal grid size for the ACCPM-PI method remains identical for all instances, since the limiting factor, here is the convergence in the oracle.

	ACCPM-PI		CPLEX		
Problem	1 proc.	mult-proc.	Simplex	IPM	+crossover
D10-2	4.8	3.4	4.8	6.4	2.5
D30-2	47.4	28.3	289	193	743
D50-2	297	187	2046	1013	>20000
D70-2	897	577	7929	1593	>20000
D10-4	16.2	6.6	14.2	21.1	8.8
D30-4	290	91	981	307	726
D50-4	1663	553	5685	1457	>20000
D10-6	35.4	15.3	35.0	22.3	31.5
D30-6	501	206	3040	458	3923
D40-6	1234	540	7498	451	> 10000
D10-12	93.9	32.5	156	68	129
D30-12	1264	420	13937	2134	> 20000

Table 3: Execution times.

Problem	calls	cuts	1 proc. time	2 proc. sp-up	3 proc. sp-up	4 proc. sp-up
D10-2	6	12	4.8	1.41	—	—
D30-2	5	10	47.4	1.67	—	—
D50-2	7	14	297	1.59	—	—
D70-2	7	14	897	1.55	—	—
D10-4	11	44	16.2	1.62	—	2.45
D30-4	10	40	290	1.90	—	3.19
D50-4	13	52	1663	1.77	—	.3.01
D10-6	16	96	35.4	1.62	2.31	—
D30-6	14	84	501	1.62	2.43	—
D40-6	15	90	1234	1.60	2.29	—
D10-12	22	264	93.9	1.64	2.29	2.89
D30-12	19	118	1264	1.66	2.25	3.01

Table 4: Speed-up of ACCPM-PI

# subproblems	ACCPM-PI	AMPL-CPLEX
2	200×200	75×75
4	200×200	55×55
6	200×200	40×40
12	200×200	30×30

Table 5: Maximal grid size solvable in a reasonable time

6 Conclusion

This paper presented a parallel implementation of a decomposition method for the computation of the solution of average cost MDPs with strong and weak interactions (or two time scales). We compared the ACCPM decomposition method, involving a policy improvement algorithm at the oracle level, with a direct LP method to solve the limit control problem. We observed (i) a sensible reduction of the execution time, (ii) a better accuracy of the policies, (iii) a sensible reduction of the RAM needed.

References

[1] Abbad, M. and J.A. Filar, Perturbation and Stability Theory for Markov Control Problems, IEEE Trans. Automatic Control, Vol. AC-37, (9), 1992, 1415-1420.

[2] Abbad, M., T. Bielecki and J.A. Filar, Algorithms for Singularly Perturbed Limiting Average Markov Control Problems, IEEE Trans. Automatic Control, Vol. AC-37, (9), 1992, 1421-1425.

[3] D. S. Atkinson and P. M. Vaidya, A cutting plane algorithm for convex programming that uses analytic centers, Mathematical Programming, Vol. 69, pp. 1–43, 1995.

[4] Bertsekas D.P., Dynamic Programming, Deterministic and Stochastic Models, Prentice Hall, 1987.

[5] Blondel V.D. and J.N. Tsitsiklis, A survey of computational complexity results in systems and control, Automatica, Vol. 36, No. 9, pp. 1249-1274, 2000.

[6] Dantzig G.B. and P. Wolfe, The Decomposition Algorithm for Linear Programming, Econometrica, Vol. 17, 1981, pp.281-296.

[7] De Ghelinck G.T., Les problèmes de décisions séquentielles, Cahier du Centre d'Etudes de Recherche Opérationnelle, 2, pp. 161-179, 1960.

[8] Delebecque F. and J.-P. Quadrat, Optimal Control of Markov Chains Admitting Strong and Weak Interactions, Automatica, Vol. 17, 1981, pp.281-296.

[9] Demel J.W., J.R. Gilbert and X.S. Li, SuperLU User's Guide, 1997.

[10] Filar J.A. and A. Haurie, Optimal Ergodic Control of Singularly Perturbed Hybrid Stochastic Systems, Lectures in Applied Mathematics, Vol. 33, AMS, 1997.

[11] Ghosh M.K., Arapostathis and S.I. Marcus, . Ergodic Control of Switching Diffusions , SIAM J. Optimization and Control, Vol . 35, N0. 6, , 1997, pp. 1952 - 1988.

[12] Goffin J.-L. and Z. Q. Luo and Y. Ye, Complexity analysis of an interior point cutting plane method for convex feasibility problems, SIAM Journal on Optimization, Vol. 6, pp. 638–652, 1996.

[13] Goffin J.-L. and J.-Ph. Vial, Convex nondifferentiable optimization: A survey focussed on the analytic center cutting plane method, Technical Report 99.02, HEC/Logilab, University of Geneva, 102 Bd Carl-Vogt, CH-1211, Switzerland, 1999.

[14] Gondzio J., R. Sarkissian and J.-P. Vial, Parallel implementation of a central decomposition method for solving large scale planning problems, To appear in Computational Optimization and Applications.

[15] Gondzio J., O. du Merle, R. Sarkissian and J.-P. Vial, ACCPM - A Library for Convex Optimization Based on an Analytic Center Cutting Plane Method, European Journal of Operational Research, Vol. 94, pp. 206–211, 1996.

[16] Haurie A., F. Moresino, J.-P. Vial, Singularly perturbed hybrid systems approximated by structured linear programs, to appear in J.Filar and Hou Zhenting Eds., Markov Processes and Controlled markov Chains, Kluwer, to appear.

[17] Kushner H. and P. Dupuis, Numerical Methods for Stochastic Control Problems in Continuous Time, Springer-Verlag, Berlin-NY, 1992.

[18] Manne A., Linear programming and sequential decisions, Management Science, 6, pp. 259-267, 1960.

[19] Moresino F., Stochastic Optimization: Numerical Methods, PhD. thesis, University of Geneva, 1999.

[20] Phillips R.G. and P. Kokotovic, A Singular Perturbation Approach to Modeling and Control of Markov Chains, IEEE Trans. Automatic Control, Vol. AC-26, (5), 1981, 1087-1094.

[21] Puterman M.L, Markov Decision Processes, J. Wiley NY, 1994.

[22] Snir M, S.W. Otto, S. Huss-Ledernan, D.W. Walker, J. Dongarra, MPI: the complete reference, MIT Press, Cambridge Mass. 1996.