On Mixed-Precision Iterative Methods and Analysis for Nearly Completely Decomposable Markov Processes

Vasileios Kalantzis, Mark S. Squillante, Chai Wah Wu
Mathematical Sciences Department, IBM Research
Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA
vkal@ibm.com, mss@us.ibm.com, cwwu@us.ibm.com

1. INTRODUCTION

Consider a discrete-time Markov process $\{X(s)\,;\, s \in \mathbb{Z}_+\}$ defined on the state space $[n] := \{1, \ldots, n\}$ with transition probability matrix $\mathbf{P}$ taking the *nearly completely decomposable* (NCD) block structural form

$$\mathbf{P} = \begin{pmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} & \cdots & \mathbf{P}_{1m} \\ \mathbf{P}_{21} & \mathbf{P}_{22} & \cdots & \mathbf{P}_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{P}_{m1} & \mathbf{P}_{m2} & \cdots & \mathbf{P}_{mm} \end{pmatrix}, \qquad (1)$$

where $\mathbf{P}_{ij} \in \mathbb{R}^{n_i \times n_j}$, $i, j \in [m]$, are nonnegative block matrices such that $\sum_{j\in[m]} \mathbf{P}_{ij}$ are stochastic, $\mathbf{P} \in \mathbb{R}^{n\times n}$ is a stochastic matrix, $n = \sum_{i\in[m]} n_i$, and the elements of $\mathbf{P}_{ij}$ are tiny relative to those of $\mathbf{P}_{ii}$, $i, j \in [m], i \neq j$; namely, $\|\mathbf{P}_{ii}\| = O(1)$ and $\|\mathbf{P}_{ij}\| = O(\epsilon)$, with $\|\cdot\|$ the spectral norm of a matrix and $\epsilon > 0$ a tiny constant. Define $\boldsymbol{\pi} := (\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_m)$, $\boldsymbol{\pi}_i := (\pi_{i1}, \ldots, \pi_{in_i})$, $\pi_{ik} := \lim_{s\to\infty} \mathbb{P}[X(s) = (\sigma_i + k)]$, $k \in [n_i]$, where $\sigma_i := \sum_{\ell=1}^{i-1} n_\ell$, $i \in [m]$. The probability vector $\boldsymbol{\pi}$ is the stationary distribution of the Markov process $\{X(s)\,;\, s \in \mathbb{Z}_+\}$. We assume this process to be irreducible and ergodic, and thus its invariant probability vector $\boldsymbol{\pi}$ exists and is uniquely determined as the solution of $\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}$ and $\boldsymbol{\pi}\mathbf{1} = \|\boldsymbol{\pi}\|_1 = 1$, where $\mathbf{1} = (1, \ldots, 1)^\top$ is the column vector containing all ones, and $\|\cdot\|_1$ is the l_1 norm of a vector.

Beyond their broad applications, NCD structural properties represent a fundamental area in the theory of Markov processes. The most efficient numerical methods for computing the stationary distribution $\boldsymbol{\pi}$ of such NCD Markov processes are based on similar forms of aggregation-disaggregation as part of iterative methods. Despite significant performance benefits, these methods can still be prohibitively expensive for highly large-scale processes and/or real-time settings. We therefore focus on the design and analysis of a general mathematical framework of numerical methods for computing the invariant vector $\boldsymbol{\pi}$ of NCD Markov processes that address the performance bottlenecks of existing methods. Our framework involves a combination of general: (1) advances in computer architectures related to mixed-precision computation – to significantly reduce computation times at the expense of inaccuracies; (2) advances in iterative approximate computing methods – to mitigate the inaccurate computations, further reduce computation times, and guarantee convergence. We derive a mathematical analysis that establishes theoretical properties of our general framework including results on approximation errors and convergence. Numerical experiments demonstrate that our general framework provides orders of magnitude improvements in computation times over existing methods. Due to space restrictions, we refer to [1] for technical details on NCD Markov processes, derivations of our algorithmic approaches and related theoretical results, proofs of our theoretical results, and a full set of references.

MAMA 2025 Stony Brook, NY, USA.

2. ALGORITHMIC SOLUTIONS

In this section we present our design of a general algorithmic framework for computing the stationary distribution $\boldsymbol{\pi}$ of NCD Markov processes. Our goal is to provide significant improvements in computational and theoretical properties over the most efficient existing numerical methods, each of which exploit aggregation-disaggregation in a similar manner with similar convergence behaviors. While our general mathematical framework can be applied to any of these numerical methods and beyond, we focus here on an application of our algorithmic framework within the context of the method due to Koury, McAllister, Stewart [2], since it is the most recent of the best existing methods. We consider two such instances of our general algorithmic framework that primarily differ in the degree of aggressiveness with which they exploit mixed-precision computation and iterative approximate computing methods in order to reduce the computational bottlenecks associated with solving systems of linear equations.

First consider the main steps of the KMS method, starting with any initial approximation $\boldsymbol{\pi}^{(0)} = (\boldsymbol{\pi}_1^{(0)}, \ldots, \boldsymbol{\pi}_m^{(0)})$. Then, for each iteration $t = 1, 2, 3, \ldots$, **Step 1** normalizes vector components of current estimate $\boldsymbol{\pi}^{(t-1)}$ of solution $\boldsymbol{\pi}$, i.e., $\hat{\boldsymbol{\pi}}_i^{(t-1)} = \frac{\boldsymbol{\pi}_i^{(t-1)}}{\|\boldsymbol{\pi}_i^{(t-1)}\|_1}$ according to $\hat{\boldsymbol{\pi}}_i := \frac{\boldsymbol{\pi}_i}{\|\boldsymbol{\pi}_i\|_1}$, $i \in [m]$. **Step 2** computes elements of aggregation matrix $\mathbf{R}^{(t-1)}$ leading up to current iteration, i.e., $\mathbf{R}_{ij}^{(t-1)} = \hat{\boldsymbol{\pi}}_i^{(t-1)}\mathbf{P}_{ij}\mathbf{1}$ according to $\mathbf{R}_{ij} := \hat{\boldsymbol{\pi}}_i\mathbf{P}_{ij}\mathbf{1}$, $i, j \in [m]$. **Step 3** obtains dominant left eigenvector $\mathbf{s}$ of $\mathbf{R}$ by computing solution of $\mathbf{s}^{(t-1)} = \mathbf{s}^{(t-1)}\mathbf{R}^{(t-1)}$ and $\mathbf{s}^{(t-1)}\mathbf{1} = \|\mathbf{s}^{(t-1)}\|_1 = 1$ according to $\mathbf{s} = \mathbf{s}\mathbf{R}$ and $\mathbf{s}\mathbf{1} = \|\mathbf{s}\|_1 = 1$. **Step 4** computes Hadamard product $\mathbf{z}^{(t)} = \mathbf{s}^{(t-1)} \odot \hat{\boldsymbol{\pi}}^{(t-1)} = (s_1^{(t-1)}\hat{\boldsymbol{\pi}}_1^{(t-1)}, \ldots, s_m^{(t-1)}\hat{\boldsymbol{\pi}}_m^{(t-1)})$. **Step 5** solves sequence of m linear systems $\boldsymbol{\pi}_i^{(t)} = \boldsymbol{\pi}_i^{(t)}\mathbf{P}_{ii} + \sum_{j<i} \mathbf{z}_j^{(t)}\mathbf{P}_{ji} + \sum_{j>i} \boldsymbol{\pi}_j^{(t)}\mathbf{P}_{ji}$ in order $i = m, \ldots, 1$, rendering blockwise estimates of components of $\boldsymbol{\pi}^{(t)}$. **Step 6** conducts test for convergence, returning current estimate $\boldsymbol{\pi}^{(t)}$ if solution is sufficiently accurate; otherwise, incrementing iteration index t and repeating iterative process of **Step 1** – **Step 6**.

The performance bottlenecks of methods such as KMS concern computing the solution of $m + 1$ systems of linear

equations in each iteration, specifically the sequence of $n_i \times n_i$ linear systems in **Step 5**, $i \in [m]$, as well the $m \times m$ linear system in **Step 3** for problems with large m. To address these bottlenecks, the first instance of our general algorithmic framework ***Alg. 3.1*** consists of replacing the standard linear solvers in **Step 5** and **Step 3** with approximate mixed-precision computing methods which involve a combination of advances in lower precision technology to reduce computation costs and advances in iterative approximate methods to further reduce computation costs with adjustments to realize full-precision results and guarantee convergence. Examples of the former advances include multi-precision arithmetic or stochastic rounding; instances of the latter include iterative refinement (IR) and Richardson iteration (RI) methods.

More precisely, as summarized in Alg. 3.1 of [1], we replace the single invocation of a full-precision linear system solver in **Step 5** and **Step 3** with a mixed-precision linear system solver based on the IR method, as a representative example. To reduce the computation costs as much as possible while still realizing full-precision results, we select the level of reduced precision based on the condition number and norm of the matrix $\mathbf{P}_{ii}$ for each linear system in **Step 5**, $i \in [m]$, and similarly for **Step 3**. Then, once the level of reduced precision has been appropriately determined, the IR method is used to solve each of the linear systems in **Step 5** and **Step 3**, generically denoted by $\mathbf{Ax} = \mathbf{b}$. At every step k of the IR method within each outer-loop iteration t, we compute the residual $\mathbf{d}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$ and then solve for $\mathbf{y}_k$ in $\mathbf{A}\mathbf{y}_k = \mathbf{d}_k$ using the reduced precision method. The resulting solution $\mathbf{y}_k$ is added to the previous estimate $\mathbf{x}_k$ to obtain $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{y}_k$, and the step index k is incremented. This process of inner-loop steps is repeated until the residual falls below the error of double-precision arithmetic [1].

Turning to the second instance of our general algorithmic framework ***Alg. 3.2***, as summarized in Alg. 3.2 of [1], we take a more aggressive approach in exploiting approximate mixed-precision computing methods to replace the standard linear solvers. In particular, **Step 5** can be computed with fewer iterations at the expense of not realizing full-precision results, which is then addressed through appropriated levels of preconditioning and iteration. We focus on **Step 5** with the understanding that the same approach can be applied in **Step 3**; alternatively, **Step 3** can employ our approximate mixed-precision computing approach in ***Alg. 3.1***. Define the block matrices $\mathbf{D}_{ii} = \mathbf{I}_{n_i} - \mathbf{P}_{ii}$, $i \in [m]$; the strictly block-lower-triangular matrices $\mathbf{L}_{ij} = \mathbf{P}_{ij}$, $i > j$, and $\mathbf{L}_{ij} = \mathbf{0}$, $i \leq j$, $i, j \in [m]$; the strictly block-upper-triangular matrices $\mathbf{U}_{ij} = \mathbf{P}_{ij}$, $i < j$, and $\mathbf{U}_{ij} = \mathbf{0}$, $i \geq j$, $i, j \in [m]$; and the block-diagonal matrix $\mathbf{D} = \text{diag}(\mathbf{D}_{11}, \ldots, \mathbf{D}_{mm})$. We can decompose the matrix $\mathbf{I}-\mathbf{P}$ as $\mathbf{I}-\mathbf{P} = \mathbf{D}-\mathbf{L}-\mathbf{U}$. Define the diagonal matrix $\mathbf{D}^{(t-1)}$ of iteration $t-1$ whose corresponding i-th principal diagonal block is set to $\mathbf{D}_{ii}^{(t-1)} = \frac{s_i^{(t-1)}}{\|\boldsymbol{\pi}_i^{(t-1)}\|_1}\mathbf{I}_{n_i}$, $i \in [m]$. From **Step 4**, we can write $\mathbf{z}^{(t)} = \boldsymbol{\pi}^{(t-1)}\mathbf{D}^{(t-1)}$ which, in combination with the linear systems in **Step 5**, leads to $\boldsymbol{\pi}^{(t)} = \mathbf{z}^{(t)}\mathbf{L}(\mathbf{D}-\mathbf{U})^{-1}$. Substituting the expression for $\mathbf{z}^{(t)}$ then yields $\boldsymbol{\pi}^{(t)} = \boldsymbol{\pi}^{(t-1)}\mathbf{D}^{(t-1)}\mathbf{L}(\mathbf{D}-\mathbf{U})^{-1}$.

With this starting point, we exploit in ***Alg. 3.2*** a combination of the above $\boldsymbol{\pi}^{(t)}$ equation together with levels of mixed-precision preconditioning and iteration for **Step 5**. More precisely, transposing both sides of the above equation yields $\boldsymbol{\pi}^{(t)\top} = (\mathbf{D}^\top - \mathbf{U}^\top)^{-1}\mathbf{L}^\top\mathbf{D}^{(t-1)}\boldsymbol{\pi}^{(t-1)\top}$, from which it is apparent that $\boldsymbol{\pi}^{(t)\top}$ can be recast as the solution of a sparse linear system with $\mathbf{D}^\top - \mathbf{U}^\top$ as its iteration matrix and $\mathbf{L}^\top\mathbf{D}^{(t-1)}\boldsymbol{\pi}^{(t-1)\top}$ as its RHS. Then our approach in **Step 5** approximately solves the linear system of this transpose equation via a fixed number of steps of the mixed-precision RI method. Hence, the preconditioned RI updates the k_t-th approximation of $\boldsymbol{\pi}^{(t)\top}$ via the fixed-point iteration $\boldsymbol{\pi}_{k_t+1}^{(t)\top} = (\mathbf{I}-\mathbf{M}^{-1}(\mathbf{D}^\top-\mathbf{U}^\top))\boldsymbol{\pi}_{k_t}^{(t)\top}+\mathbf{M}^{-1}\mathbf{L}^\top\mathbf{D}^{(t-1)}\boldsymbol{\pi}^{(t-1)\top}$, where k_t denotes the number of mixed-precision RI steps in the t-th outer-loop iteration of ***Alg. 3.2*** and the matrix $\mathbf{M}^{-1}$ denotes the preconditioner of the iterative method. For example, if $\mathbf{M} \equiv \mathbf{D}^\top$, i.e., block-Jacobi preconditioning, the iteration matrix in the above fixed-point equation becomes equal to $\mathbf{D}^{-\top}\mathbf{U}^\top$. We consider $\mathbf{M}$ to be the product of the LU factors of $\mathbf{D}^\top$ obtained using reduced precision.

3. MATHEMATICAL ANALYSIS

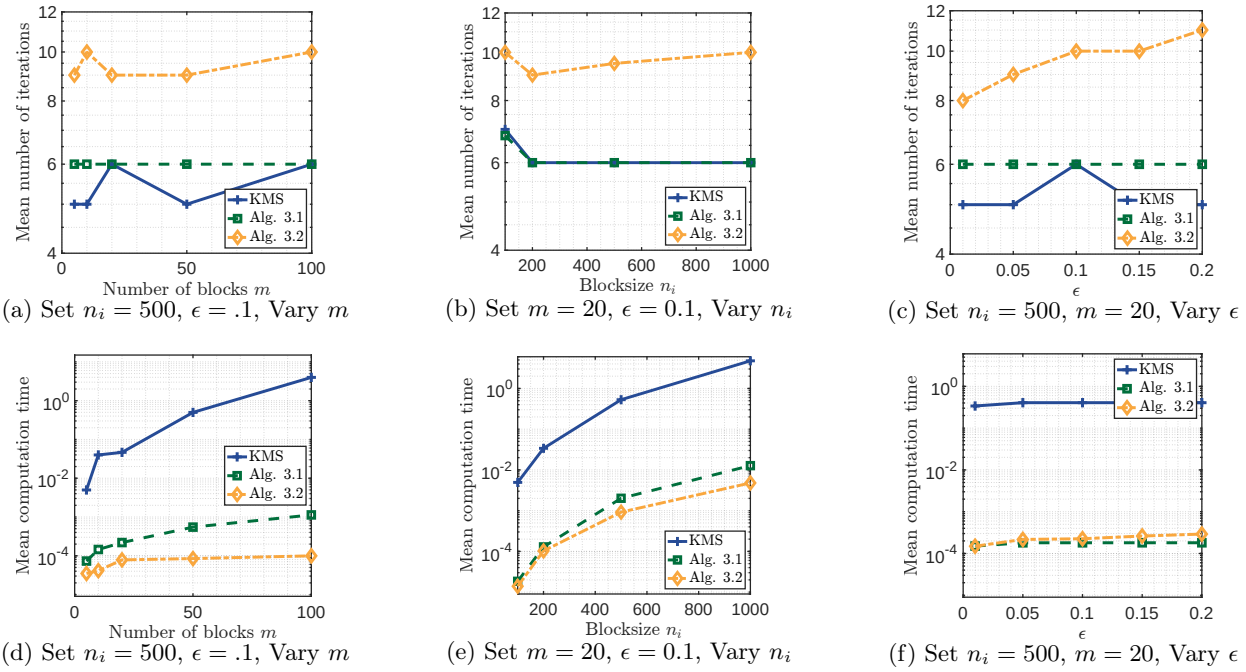
We now turn to derive a mathematical analysis of our general algorithmic framework, first summarizing our main theoretical results for both instances and then summarizing our performance analysis of the computational improvements. Our main theoretical results establish that ***Alg. 3.1*** is guaranteed to converge with an approximation error $\|\boldsymbol{\pi}^{(t)} - \boldsymbol{\pi}\|$ decreasing by a factor of $O(\epsilon)$ at each iteration t; and that the inner-loop use of IR is guaranteed to converge linearly with a decreasing factor which is typically much faster than the decreasing factor of $O(\epsilon)$ for the outer-loop convergence.

THEOREM 3.1. ***Alg. 3.1*** *converges with an approximation error in the solution $\boldsymbol{\pi}^{(t)}$ at each iteration t that decreases by a factor of $O(\epsilon)$. The mixed-precision IR method for computing the solution of the linear system $\mathbf{Ax} = \mathbf{b}$ in* ***Steps 3*** *and* **5** *of* ***Alg. 3.1*** *converges linearly with an approximation error that decreases by a factor of $O(\kappa(\mathbf{A}))$ in each iteration.*

Our main theoretical results also establish that ***Alg. 3.2*** is guaranteed to converge with an error $\|\boldsymbol{\pi}^{(t)} - \boldsymbol{\pi}\|$ decreasing in regard to a tradeoff between an increased number of outer-loop iterations t with fewer RI steps k_t in **Step 5** and a decreased computational complexity of the RI method for smaller k_t. We derive numerical linear algebraic methods to address such well-known tradeoffs. By starting with an inexact application of the RI method and increasing the number of RI steps k_t with the number of outer-loop iterations t, we avoid over-solving in the earlier stages of **Step 5** when the estimate $\boldsymbol{\pi}^{(t)}$ is far from $\boldsymbol{\pi}$; e.g., the tolerance up to which we solve the linear system in **Step 5** follows a geometric criterion such as γ^t or $\frac{\rho}{(t+1)^\theta}$ for some $\gamma, \rho \in (0,1)$ and $\theta > 1$.

THEOREM 3.2. *Suppose* ***Alg. 3.2*** *with exact linear system solutions in* ***Step*** **5** *provides a sequence $\{\boldsymbol{\pi}^{(t)}\}$ such that $\|\boldsymbol{\pi}^{(t)} - \boldsymbol{\pi}\| < \|\boldsymbol{\pi}^{(t-1)} - \boldsymbol{\pi}\|$, $t > 1$. Then, the sequence $\{\overline{\boldsymbol{\pi}}^{(t)}\}$ produced by replacing these exact linear system solutions in* ***Step*** **5** *of* ***Alg. 3.2*** *with k_t steps of the RI method converges to the same limit as $\{\boldsymbol{\pi}^{(t)}\}$, provided that k_t increases (slowly) with the number of outer-loop iterations t.*

The performance tradeoff at the heart of our general algorithmic framework concerns, on the one hand, the significant reductions in execution times afforded by mixed-precision computation at the expense of inaccuracies in the results and, on the other hand, the ability to mitigate inaccurate computations, further reduce execution times, and guarantee convergence afforded by iterative approximate computing methods. Typically, there is a factor of 2× reduction in computation times with respect to the number of operations per



(a) Set $n_i = 500$, $\epsilon = .1$, Vary m (b) Set $m = 20$, $\epsilon = 0.1$, Vary n_i (c) Set $n_i = 500$, $m = 20$, Vary ϵ

(d) Set $n_i = 500$, $\epsilon = .1$, Vary m (e) Set $m = 20$, $\epsilon = 0.1$, Vary n_i (f) Set $n_i = 500$, $m = 20$, Vary ϵ

Figure 1: Performance of our Algorithm 3.1 and Algorithm 3.2 and the KMS baseline for diagonal matrices $\mathbf{P}_{ii}$ with dimension n_i, number of blocks m, and off-diagonal matrices $\|\mathbf{P}_{ij}\| = O(\epsilon)$, averaged over 10 independent trial runs.

second (OPS) going from 64-bit precision to 32-bit precision, and orders of magnitude factors of reduction going to 16-bit precision and to 8-bit precision. As established by our theoretical results above and our empirical results below, the cost of any additional iterations is often minimal in comparison with the significant reductions in each iteration, and thus our general algorithmic framework provides tremendous performance improvements over the most efficient existing methods. In addition, our use of mixed-precision makes it possible to handle much larger linear systems before hitting the memory bandwidth limitations of today's advanced processors, in which performance is significantly reduced from the available number of OPS for sufficiently large problems relative to the available memory of the processor, and is instead dictated by the memory bandwidth of the processor architecture. By exploiting reduced precision computation, our general algorithmic framework further enables us to handle significantly larger linear systems in each iteration at the computational performance afforded by the processor architecture, which is well beyond what is possible with existing methods whose performance becomes cache/memory-bandwidth limited for much smaller problems.

4. NUMERICAL EXPERIMENTS

In this section we present a representative sample of numerical experiments for solving NCD Markov processes that support our theoretical results and empirically evaluate our general algorithmic framework, demonstrating that our framework exhibits relatively little or no increase in the number of outer-loop iterations and orders of magnitude improvements in the computation time over the most efficient KMS baseline. Fig. 1 presents the mean number of iterations and the mean computation time (in seconds) for computing the stationary distribution of NCD Markov processes with ***Alg. 3.1***, ***Alg. 3.2***, and the KMS baseline, averaged over ten independent experimental trial runs. Specifically, Fig. 1(a),1(d) fixes the dimensions $n_i = 500$ of the diagonal block matrices $\mathbf{P}_{ii}$ and the magnitude $\epsilon = 0.1$ of transitions in the off-diagonal block matrices $\|\mathbf{P}_{ij}\| = O(\epsilon)$, $i \neq j$, while varying the number $m \in \{5, 10, 20, 50, 100\}$ of diagonal block matrices; Fig. 1(b),1(e) fixes $m = 20$ and $\epsilon = 0.1$, while varying $n_i \in \{100, 200, 500, 1000\}$; Fig. 1(c),1(f) fixes $n_i = 500$ and $m = 20$, while varying $\epsilon \in \{0.01, 0.05, 0.1, 0.15, 0.2\}$. We first observe that the mean number of iterations under ***Alg. 3.1*** are essentially identical to that of the baseline KMS, with the mean number of iterations under ***Alg. 3.2*** only somewhat higher requiring a few additional iterations (though of lower computational complexity). The mean number of iterations as a function of the varying parameter remains relatively flat for all algorithms, except for ***Alg. 3.2*** requiring slightly more outer-loop iterations as ϵ increases, all as expected due to the properties of the varying parameters. We further observe that both instances of our algorithmic framework provide orders of magnitude reduction in the mean computation time, linearly increasing with m and n_i and remaining consistent with increasing ϵ. Our ***Alg. 3.2*** instance provides the lowest mean computation times, although the differences with ***Alg. 3.1*** are relatively small with respect to n_i and ϵ due to the nature of the block-Jacobi preconditioner.

5. REFERENCES

[1] V. Kalantzis, M.S. Squillante, C.W. Wu. On mixed-precision iterative methods and analysis for nearly completely decomposable Markov processes. arXiv:2504.06378, 2025.

[2] J. Koury, D. McAllister, W. Stewart. Iterative methods for computing stationary distributions of nearly completely decomposable Markov chains. *SIAM J.DMA*, 5:164–186, 1984.