An Interior Point Approach to Large Games of Incomplete Information

François Pays
francois.pays@gmail.com

Abstract

Since their discovery 30 years ago, interior point methods deliver the most competitive algorithms for large scale optimization. Surprisingly, even when games of incomplete information can be formulated as a linear program, interior point methods have been discarded in favor of usually less attractive methods. This paper describes how specialized interior point methods can also scale to large games.

Introduction

Games of perfect information, such as chess, are nowadays commonly solved using backward induction of the game tree. Figuring out how to solve efficiently incomplete information games, such as poker, is an ongoing matter. See (Rubin and Watson 2011) for a recent survey.

We consider two-person zero-sum sequential games with incomplete information. We will also suppose perfect recall although this requirement will be partially relaxed later. Our goal is to solve either the original game or, as it is often the case, smaller tractable abstractions.

Optimal strategies are solutions of the following min-max problem

$$\max_{x \in \mathcal{X}} \min_{y \in \mathcal{Y}} x^T A y \;=\; \min_{y \in \mathcal{Y}} \max_{x \in \mathcal{X}} x^T A y. \qquad (1)$$

The standard normal-form approach involves exponential blow-up and is simply not practical. However, (Koller, Megiddo, and Von Stengel 1994) reinvented *sequence form* (Romanovskii 1962) and proved that optimal strategies are solutions of a linear program (LP) whose size, in sparse representation, is linear in the size of the game tree.

But in practice, the LP approach is not often used when it comes to large games, and current popular techniques follow some other approaches. Among them are:

- The "Conterfactual Regret Minimization" (CFR) is a very powerful algorithm based on regret minimization and fictitious play (Zinkevich et al. 2007). Although it finds approximations of the game in $\mathcal{O}(1/\epsilon^2)$ iterations, it is as of today the preferred algorithm with many refinements.
- The "Excessive Gap Technique" (EGT) is also an established approach. It is a gradient-based algorithm based on Nesterov's first-order smoothing method (Gilpin et al. 2007). It needs $\mathcal{O}(1/\epsilon)$ iterations, later refined to $\mathcal{O}(\ln(1/\epsilon))$.

Nowadays, interior point methods (IPMs) are fully developed and generally the favorite numerical approach to large scale optimization. They deliver robust algorithms guaranteed to find solutions in $\mathcal{O}(\ln(1/\epsilon))$ iterations. While CFR and EGT have proven their practical efficiency, a credible interior point approach to this problem would be very compelling.

In publications dedicated to solving large games, the IPM approach is often mentioned but always dismissed because of excessive computational costs and memory requirements. In the past, attempts have been made to use IPMs to solve the equivalent LP, but numerical results using off-the-shelf commercial software suggested that they could not scale to tackle large games. The factorization of the inner Newton system is seen as a major blocking point. (Rubin and Watson 2011; Sandholm 2010; Gilpin et al. 2007; Hoda, Gilpin, and Pena 2007).

Indeed, for a long time, it was thought that the ill-conditioning of the Newton system precluded the use of iterative methods. But in the late 90s, as optimization problems got larger, iterative methods for IPMs started to be actively researched and proved ultimately to be reliable when appropriately preconditioned. Today, iterative methods for IPMs are an active field of research for solving truly large scale linear and quadratic programming problems. See (Gondzio 2012a; 2012b) for a recent survey of the IPMs as well as a matrix-free design.

The key idea of the approach presented in this paper is to apply specific projected Krylov methods along with implicit preconditioning to the sequence of barrier subproblems. Replacing direct algebra with iterative methods evades the bottleneck that faced previous IPM attempts. Conversely, we concentrate our efforts on reducing the spectral condition of the preconditioned system matrix. We show that the resulting IPM has minimal memory requirements while maintaining a very good efficiency.

Interior Point Formulation

First Order Conditions

Using the Koller's notation of a sequence form game (Koller, Megiddo, and Von Stengel 1994), the Karush-Kuhn-Tucker (KKT) conditions are

$$\begin{aligned} &E^T\lambda_x + s_x + Ay = 0, \\ &F^T\lambda_y + s_y - A^T x = 0, \\ &Ex - e = 0, \quad Fy - f = 0, \\ &XS_x\mathbb{1} = 0, \quad YS_y\mathbb{1} = 0, \\ &x, y, s_x, s_y \geq 0. \end{aligned} \tag{2}$$

- x and y are the searched vectors of realization weights for respective players, of sizes n_x and n_y.
- A, $n_x \times n_y$, is the *payoff matrix*.
- E, $m_x \times n_x$ and F, $m_y \times n_y$, are the full rank *constraint matrices* with $m_x < n_x$ and $m_y < n_y$.
- λ_x and λ_y, of sizes m_x and m_y, are the *dual variables*.
- s_x and s_y, of sizes n_x and n_y, are the *dual slacks*.
- X, Y, S_x, S_y are the diagonal matrices of vectors x, y, s_x and s_y.
- I is the identity and $\mathbb{1}$ is the unit vector.

Let us restates the KKT conditions using

$$\mathcal{A} = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}, \quad C = \begin{bmatrix} E & 0 \\ 0 & -F \end{bmatrix}, \qquad c = \begin{bmatrix} e \\ -f \end{bmatrix},$$
$$S = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}, \quad J = \begin{bmatrix} I & 0 \\ 0 & -I \end{bmatrix}, \qquad z = \begin{bmatrix} x \\ y \end{bmatrix},$$
$$Z = \begin{bmatrix} X & 0 \\ 0 & Y \end{bmatrix}, \quad \lambda = \begin{bmatrix} \lambda_x \\ \lambda_y \end{bmatrix}, \qquad s = \begin{bmatrix} s_x \\ s_y \end{bmatrix},$$

with

- s and z are size $n = n_x + n_y$,
- λ is size $m = m_x + m_y$,
- $\mathcal{A}$ is $n \times n$ and C is $m \times n$.

We obtain

$$\begin{aligned} &J\mathcal{A}z + s + JC^T\lambda = 0, \\ &Cz = c, \\ &ZS\mathbb{1} = 0, \\ &Z, S \succeq 0. \end{aligned} \tag{3}$$

Observe that (3) is similar to the optimality conditions of a quadratic programming problem, except that we have the skew-symmetric $-J\mathcal{A}$ in place of a positive-definite Hessian.

Nevertheless, as long as we are able to solve the forthcoming Newton system, nothing prevents us to successfully make use of the barrier method. See (Boyd and Vandenberghe 2004; Ghosh and Boyd 2003) for the derivation of the barrier method for convex-concave games.

Newton System

Proceeding with the infeasible method (Wright 1997), $\mu = \frac{z^T s}{n}$ is the duality measure and $\sigma \in [0, 1]$ is the centering parameter. The residual is

$$r = \begin{bmatrix} \mathcal{A}z + Js + C^T\lambda \\ Cz - c \\ ZS\mathbb{1} - \sigma\mu\mathbb{1} \end{bmatrix}. \tag{4}$$

Therefore the Newton equations to derive the search directions are

$$\begin{bmatrix} \mathcal{A} & C^T & J \\ C & 0 & 0 \\ S & 0 & Z \end{bmatrix} \begin{bmatrix} \Delta_z \\ \Delta_\lambda \\ \Delta_s \end{bmatrix} = -r. \tag{5}$$

Eliminating Δ_s we obtain the *augmented* system form of the Newton equations:

$$\begin{bmatrix} \mathcal{A} - J\Theta & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \Delta_z \\ \Delta_\lambda \end{bmatrix} = -r_a, \tag{6}$$

using the notation

$$\Theta = \text{diag}(\theta) = Z^{-1}S = \text{diag}(z_i^{-1}s_i), i = 1, ..., n.$$

This is the saddle-point system that we need to solve efficiently at every interior point step. (Benzi, Golub, and Liesen 2005) offers a very exhaustive review of existing numerical methods for general saddle-point systems.

Note that, in the quadratic case, we would have $-Q - \Theta$ in the top-left block, which is a much more convenient negative-definite block. This property would greatly expand the choice of methods for solving the system, in a variety of forms.

In our case, $\mathcal{A} - J\Theta$ is a legitimate cause of concern. Not only the block is highly indefinite, but it is also increasingly ill-conditioned as the algorithm converges to the solution. Since the underlying problem is almost certainly degenerate (as most of real world problems are), we can anticipate the condition number of the system to get as large as $\mathcal{O}(\mu^{-2})$ (Gondzio 2012b), similarly to the quadratic case.

On the bright side, C is full rank and, as we will see later, has a special structure than we can exploit. Finally, while the system is very large, it is also very sparse.

Naturally, smaller systems would allow *dense direct* methods such as the Gaussian elimination, but on larger systems, their cubic complexity and memory usage become quickly prohibitive. On larger systems, the *sparse direct* algorithms such as the indefinite factorization package MA57 of the HSL library may be contemplated, but yet again, up to a certain size.

On very large systems, the problem will occupy so much space in memory, in sparse representation, that it cannot be modified at all. This is the exclusive domain of indirect, or iterative methods. Ultimately, in *matrix-free* regime, the coefficient matrix is not even stored in memory.

Hopefully, our system is eligible to a recent addition to the arsenal of indirect numerical methods: the *projected* Krylov subspace methods combined with appropriate *implicit* constraint preconditioners.

Exact Regularization

Prior to introduce the Krylov methods, we need to regularize the system. Regularization is a common and important safeguard to preserve the conditioning of the system and to

protect the convergence of the Krylov solvers. We employ an exact primal regularization. See (Friedlander and Orban 2012) for details. The actual positive impacts on the convergence of the Krylov methods will be uncovered later.

The regularization parameter γ is a scalar typically in the range of 10^{-3} to 10^{-2}. Our system is modified in the following way:

$$\Theta = Z^{-1}S + \gamma^2 I. \tag{7}$$

Notice the *exact* nature of this regularization: the system is modified similarly to a Tikhonov's regularization, but the residual is left untouched. It implies that we expect to recover an exact (i.e not perturbed) solution of our problem. However, the regularization has to be carefully performed: while the regularization benefits to the system conditioning, it also perturbs the derived search directions. Ultimately, a constant regularization parameter *limits* the maximum attainable accuracy of the path-following method. While we may think of more sophisticated strategies, the simplest of them is to precompute a suitable parameter value respective to the target accuracy.

We can immediately observe that, thanks to the regularization, we now have $\gamma^2 < \theta_i$.

Projected Krylov Methods

(Keller, Gould, and Wathen 2000) proposed a conjugate gradient-like algorithm to solve the saddle point system

$$\begin{bmatrix} H & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} c \\ d \end{bmatrix}, \tag{8}$$

where H is positive-definite on the nullspace of C, using a constraint preconditioner

$$\mathcal{G} = \begin{bmatrix} G & C^T \\ C & 0 \end{bmatrix}, \tag{9}$$

where G approximates H and $\mathcal{G}$ is easily solved.

The *projected preconditioned conjugate gradient* (PPCG) operates on a reduced linear system in the nullspace of C using any kind of constraint preconditioner. In place of the constraint preconditioner, (Dollar and Wathen 2004) proposed a class of *incomplete factorizations* for saddle point problems, based upon earlier work by Schilders. More recently, (Gould, Orban, and Rees 2013) extended this approach to a broader range of Krylov methods such as MINRES for possibly indefinite H.

Constraint Structure

The special structure of C is of importance and need to be exploited.

In sequence form, strategy probabilities are replaced by much smaller strategy supports called *realization plans*. Each player's vector (x or y) represents the sequence of realization weights of every choice (or move) at every *information set*. See (Koller, Megiddo, and Von Stengel 1996) for details. E and F are the *constraint matrices*. The equalities $Ex = e$ and $Fy = f$ merely verify that x and y are valid realization plans with respect to the game.

Examining E: the matrix has as many rows as the first player has information sets, and as many columns as the player has choices in the game. An information set is merely a distinct state of information from a player's perspective where he is expected to make a choice. Every row of E has:

- one coefficient -1 corresponding to the move from which the information originates,
- one or many coefficient 1 corresponding to the moves made from this information set.

$$E = \left(\underbrace{\begin{array}{cccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots \\ -1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & \cdots \\ -1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & \cdots \\ \vdots & & & & & & & & \end{array}}_{\text{move sets}} \right) \Bigg\} \text{info sets} \tag{10}$$

Because the game is sequential, the following important property applies: $E_{k,j} = 0 \;\forall\; (i,j,k)$ such that: $j \in movesets_i$ and $k < i$. Consequently:

Proposition 1. *A sequence form constraint matrix E can be split into $[E_1 \quad E_2]$ where E_1 is unit lower triangular using a column permutation.*

Construction. For every infoset, we select a move from the moveset. That is, for every row, we select a column to be permuted to E_1 and we leave the remaining columns to E_2. This process correctly partitions the column set and leaves E_1 unit lower triangular. □

We call P_x and P_y the permutation matrices such that

$$EP_x = \begin{bmatrix} E_1 & E_2 \end{bmatrix} \text{ and } FP_y = \begin{bmatrix} F_1 & F_2 \end{bmatrix}.$$

It follows immediately that C can also be split into $[C_1 \quad C_2]$ with $C_1 J$ unit lower triangular. P is the corresponding permutation of columns of z such that

$$\widetilde{C} = CP = \begin{bmatrix} C_1 & C_2 \end{bmatrix} = \begin{bmatrix} E_1 & 0 & E_2 & 0 \\ 0 & -F_1 & 0 & -F_2 \end{bmatrix}. \tag{11}$$

Schilders's Factorization

Using the permutation matrix P defined is the previous section, let us split Θ:

$$\widetilde{\Theta} = P^T \Theta P = \begin{bmatrix} \Theta_1 & 0 \\ 0 & \Theta_2 \end{bmatrix} = \begin{bmatrix} \Theta_{1_x} & 0 & 0 & 0 \\ 0 & \Theta_{1_y} & 0 & 0 \\ 0 & 0 & \Theta_{2_x} & 0 \\ 0 & 0 & 0 & \Theta_{2_y} \end{bmatrix},$$

and the payoff matrix:

$$P_x^T A P_y = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}.$$

Let us split $\mathcal{A}$ accordingly:

$$\widetilde{\mathcal{A}} = P^T \mathcal{A} P = \begin{bmatrix} \mathcal{A}_{11} & \mathcal{A}_{12} \\ \mathcal{A}_{12}^T & \mathcal{A}_{22} \end{bmatrix}.$$

We keep in mind the special 2×2 block structure of $\mathcal{A}_{i,j}$:

$$\mathcal{A}_{11} = \begin{bmatrix} 0 & A_{11} \\ A_{11}^T & 0 \end{bmatrix}, \qquad \mathcal{A}_{22} = \begin{bmatrix} 0 & A_{22} \\ A_{22}^T & 0 \end{bmatrix}, \qquad (12)$$
$$\mathcal{A}_{12} = \begin{bmatrix} 0 & A_{12} \\ A_{21}^T & 0 \end{bmatrix}.$$

Let $\mathcal{P}$ be the permutation matrix of $(z^T \lambda^T)^T$:

$$\mathcal{P} = \begin{bmatrix} P & 0 \\ 0 & I \end{bmatrix}.$$

Note that, in the permuted space, we conserve the notation J for a matrix with the same structure, appropriately sized and partitioned between the 'x' and 'y' subspaces.

In the $\mathcal{P}$-permuted space, matrix $\mathcal{H}$ of system (6) becomes

$$\widetilde{\mathcal{H}} = \mathcal{P}^T \mathcal{H} \mathcal{P} = \begin{bmatrix} \mathcal{A}_{11} - J\Theta_1 & \mathcal{A}_{12} & C_1^T \\ \mathcal{A}_{12}^T & \mathcal{A}_{22} - J\Theta_2 & C_2^T \\ C_1 & C_2 & 0 \end{bmatrix}. \qquad (13)$$

That is

$$\widetilde{\mathcal{H}} = \begin{bmatrix} \widetilde{H} & \widetilde{C}^T \\ \widetilde{C} & 0 \end{bmatrix},$$

with

$$\widetilde{H} = \begin{bmatrix} \mathcal{A}_{11} - J\Theta_1 & \mathcal{A}_{12} \\ \mathcal{A}_{12}^T & \mathcal{A}_{22} - J\Theta_2 \end{bmatrix}.$$

Similarly to (Dollar and Wathen 2004), we consider the following Schilders's factorization $Q^T B Q$ where

$$Q = \begin{bmatrix} 0 & 0 & I \\ 0 & I & 0 \\ C_1 & C_2 & 0 \end{bmatrix} \text{ and } B = \begin{bmatrix} 0 & 0 & I \\ 0 & D\Theta_2 & 0 \\ I & 0 & 0 \end{bmatrix}. \qquad (14)$$

The diagonal inversible matrix D will be either I or $-J$ in the next sections, depending on the Krylov method considered. The preconditioner in the permuted space is

$$\widetilde{\mathcal{G}} = Q^T B Q = \begin{bmatrix} 0 & 0 & C_1^T \\ 0 & D\Theta_2 & C_2^T \\ C_1 & C_2 & 0 \end{bmatrix}. \qquad (15)$$

That is

$$\widetilde{\mathcal{G}} = \begin{bmatrix} \widetilde{G} & \widetilde{C}^T \\ \widetilde{C} & 0 \end{bmatrix} \text{ with } \widetilde{G} = \begin{bmatrix} 0 & 0 \\ 0 & D\Theta_2 \end{bmatrix}.$$

Naturally, our preconditioner $\mathcal{G} = \mathcal{P}\widetilde{\mathcal{G}}\mathcal{P}^T$ is never explicitly formed and kept in the factored form

$$\mathcal{G} = \mathcal{P} Q^T B Q \mathcal{P}^T. \qquad (16)$$

At every interior point step, the matrix C is *implicitly* split using the construction algorithm. Only the permutation matrices are actually constructed, that is pairs of one-dimensional array of indices, and no other data structures need to be created. The projected Krylov method requires, from its preconditioner, the solution of a linear system with $\mathcal{G}$ and a different right hand side at every Krylov step. The solution of this linear system is obtained by forward/backward substitution in (16).

Unlike direct methods such as the Cholesky factorization, the memory requirements of short-recurrence iterative methods are minimal. The only significant memory usage is the storage of the matrices A, E and F in sparse representation, or even more condensed formats when available.

Spectral Analysis

The convergence of the Krylov methods depends on the spread of the eigenvalues of the preconditioned system matrix $\mathcal{G}^{-1}\mathcal{H}$. (Keller, Gould, and Wathen 2000) proved that the similar matrix

$$\widetilde{\mathcal{G}}^{-1}\widetilde{\mathcal{H}} \qquad (17)$$

has $2m$ eigenvalues of value 1, and $n - m$ eigenvalues solutions of the generalized eigenvalue problem

$$N^T \widetilde{H} N u = \nu N^T \widetilde{G} N u, \qquad (18)$$

where N is a nullspace basis for $\widetilde{C}$. Just like (Dollar and Wathen 2004), we use as a nullspace basis for $\widetilde{C}$

$$N = \begin{bmatrix} R \\ I \end{bmatrix},$$

with

$$R = \begin{bmatrix} -C_1^{-1} C_2 \end{bmatrix} = \begin{bmatrix} R_x & 0 \\ 0 & R_y \end{bmatrix} = \begin{bmatrix} -E_1^{-1} E_2 & 0 \\ 0 & F_1^{-1} F_2 \end{bmatrix}.$$

It follows

$$N^T \widetilde{H} N = N^T \widetilde{\mathcal{A}} N - R^T J \Theta_1 R - J\Theta_2,$$
$$N^T \widetilde{G} N = D\Theta_2.$$

Since D is either I of $-J$, the generalized eigenvalue problem (18) is equivalent to the eigenvalue problem for matrix $D^{-1}K$ with

$$K = \Theta_2^{-1/2} N^T \widetilde{\mathcal{H}} N \Theta_2^{-1/2}$$
$$= \Theta_2^{-1/2} N^T \widetilde{\mathcal{A}} N \Theta_2^{-1/2} - \Theta_2^{-1/2} R^T J \Theta_1 R \Theta_2^{-1/2} - J.$$

Recall the 2×2 block structure of $\mathcal{A}_{ij}$ (12) and observe that

$$K = \begin{bmatrix} -K_x & K_s^T \\ K_s & K_y \end{bmatrix} = K_{diag} + K_{off}, \qquad (19)$$

with

$$K_{diag} = \begin{bmatrix} -K_x & 0 \\ 0 & K_y \end{bmatrix} = -J - J\Theta_2^{-1/2} R^T \Theta_1 R \Theta_2^{-1/2},$$
$$K_{off} = \begin{bmatrix} 0 & K_s^T \\ K_s & 0 \end{bmatrix} = \Theta_2^{-1/2} N^T \widetilde{\mathcal{A}} N \Theta_2^{-1/2}.$$

Explicitly:

$$K_x = I + \Theta_{2_x}^{-1/2} R_x^T \Theta_{1_x} R_x \Theta_{2_x}^{-1/2} = I + L_x,$$
$$K_y = I + \Theta_{2_y}^{-1/2} R_y^T \Theta_{1_y} R_y \Theta_{2_y}^{-1/2} = I + L_y,$$
$$K_s = \Theta_{2_y}^{-1/2} (R_y^T A_{11}^T R_x + A_{12}^T R_x + R_y^T A_{21}^T + A_{22}^T) \Theta_{2_x}^{-1/2}.$$

Split Choice From a preconditioning viewpoint, we want to promote the identity in front of L_x, L_y and K_s. Since the constraints split process leaves a choice of column at every row, we choose θ_2 to contain the *largest* possible elements and, conversely, θ_1 the *smallest*. To achieve this goal, we choose for C_1, at each row i of C, the column

$$\arg \max_{j \in moveset_i} z_j^{-1} s_j.$$

Naturally, we cannot expect to have all the elements of θ_1 smaller than the elements of θ_2. But we expect the geometric mean of θ_1 to be much lower (by an order of magnitude) to the geometric mean of θ_2.

Let us complete the spectral analysis depending on the choice for D (either I or $-J$).

Projected MINRES

Let us consider the preconditioner when $D = I$. The preconditioner in the permuted space is

$$\widetilde{\mathcal{G}} = Q^T B_I Q = \begin{bmatrix} 0 & 0 & C_1^T \\ 0 & \Theta_2 & C_2^T \\ C_1 & C_2 & 0 \end{bmatrix}. \tag{20}$$

First and foremost, we observe that $N^T \widetilde{G} N = \Theta_2$ is positive-definite, thus $\widetilde{\mathcal{G}}$ is positive-definite on the nullspace of $\widetilde{C}$. Consequently we are permitted to use the projected version on MINRES (Paige and Saunders 1975) which is a Krylov subspace method based on Lanczos tridiagonalization for solving symmetric indefinite systems. The projected variant using a constraint preconditioner is analyzed in (Gould, Orban, and Rees 2013).

Proposition 2. *The matrix of the preconditioned system* (17)*, when $D = I$, has $2m$ eigenvalues of value* 1*, $n_x - m_x$ negative and $n_y - m_y$ positive. Any eigenvalue ν satisfies*

$$1 \leq |\nu| \leq 1 + \Omega^2 \frac{\|A\| + \theta_{1\,max}}{\theta_{2\,min}}, \tag{21}$$

where Ω is a constant that depends on E and F.

Proof.

Lower bound The remaining $n - m$ eigenvalues of the preconditioned system matrix (17) are the eigenvalues of K. Observe that L_x and L_y are square positive-definite. The eigenvalue equation is

$$(K - \nu I)u = \begin{bmatrix} -(1+\nu)I - L_x & K_s^T \\ K_s & (1-\nu)I + L_y \end{bmatrix} u = 0.$$

Suppose $|\nu| < 1$, $L_x + (1+\nu)I$ is positive-definite and the Schur complement $(1-\nu)I + L_y + K_s^T (L_x + (1+\nu)I)^{-1} K_s$ cannot be singular, since it is positive-definite. Therefore ν eigenvalue of K implies $1 \leq |\nu|$.

Upper bound Since K is symmetric, the spectral radius equals the 2-norm: $\rho(K) = \max_i |\nu_i| = \|K\|$. We obtain

$$\|K\| \leq \|K_{diag}\| + \|K_{off}\| \leq 1 + \|R\|^2 \frac{\|A\| + \theta_{1\,max}}{\theta_{2\,min}}.$$

Considering the structure of C, C_1 and C_2, $\|R\| \leq \max(\left\|E_1^{-1}\right\| \|E_2\|, \left\|F_1^{-1}\right\| \|F_2\|)$ depends marginally on the choice of the split $[C_1 C_2]$. We name Ω the constant that bounds $\|R\|$ over the worse split. Although we acknowledge that we cannot practically compute this value, Ω is likely in the range of $\|E\|$ and $\|F\|$.

Inertia $-K_x$ is negative-definite and has rank $n_x - m_x$. The Schur complement $K_y + K_s^T K_x^{-1} K_s$ is positive-definite and has rank $n_y - m_y$. Sylvester's Law of Inertia implies that K has $n_x - m_x$ negative eigenvalues and $n_y - m_y$ positive eigenvalues.

□

Convergence Let us examine K_s, K_x and K_y as the IPM converges to the solution. $z_i^{-1} s_i$ can have a very wide range of values: we expect the lowest elements to be $o(\mu)$ and the larger elements to be $\mathcal{O}(\mu^{-1})$. After the split, we expect θ_2 to contain slightly larger elements than θ_1, in geometric mean. Finally, recall that θ_2 cannot be smaller than γ^2.

Considering the range of values of θ_1 and θ_2, L_x and L_y perturb the identity only when θ_{1i} is large and θ_{2j} is small (but never smaller than γ^2). Therefore we expect K_x and K_y to have a large number of eigenvalues into the upper vicinity of 1, and a smaller set of possibly large ($\mathcal{O}(\mu^{-1})$) eigenvalues. Finally, we expect the coefficients of K_s to be bounded and highly sparse, since they vanish everywhere except where θ_{2_x} and θ_{2_y} are not too large.

Consequently, we expect a good conditioning of the preconditioned system and a very large number of eigenvalues in the vicinity of 1 and -1.

Projected Conjugate Gradient

It is very tempting to consider the preconditioner resulting from the choice $D = -J$. Unfortunately, at first sight, the resulting preconditioner is not applicable to the projected conjugate gradient.

However, our numerical experiments indicate that CG preconditioned in this manner, not only works, but significantly outperforms MINRES. These excellent practical results suggest a closer analysis.

The preconditioner in the permuted space is

$$\widetilde{\mathcal{G}} = Q^T B_{-J} Q = \begin{bmatrix} 0 & 0 & C_1^T \\ 0 & -J\Theta_2 & C_2^T \\ C_1 & C_2 & 0 \end{bmatrix}. \tag{22}$$

The preconditioner central block is an exact replication of the diagonal of $\widetilde{\mathcal{H}}$ (13), but is indefinite on the nullspace of $\widetilde{C}$.

The remaining $n - m$ eigenvalues of the preconditioned system matrix (17) are the eigenvalues of

$$K' = -JK = \begin{bmatrix} K_x & -K_s^T \\ K_s & K_y \end{bmatrix}. \tag{23}$$

Our first observation is that the matrix K' is unsymmetric. It follows immediately that CG is not well defined on the corresponding system. Indeed, the conjugate gradient will be unable to build an orthogonal basis (Faber and Manteuffel 1984).

The second observation is that K' is (unsymmetric) *positive-definite* in the sense that $u^T K' u > 0$ for all $u \neq 0$. This is not a sufficient condition for the conjugate gradient to be applicable, but this means that no breakdown will ever happen during the course of the method. Despite K' positive-definite, $u \to u^T K' u$ is not a proper inner product since K' is not symmetric.

Examining the numerical range of K' (containing all its eigenvalues), we learn immediately that K' is *positive stable*. Indeed, the eigenvalues of K' have real parts in the interval $[1, \max(\|K_x\|, \|K_y\|)]$, and imaginary parts in the interval $[-\|K_s\|, \|K_s\|]$.

Convergence In practice, it is common to run the Hestenes-Stiefel conjugate gradient to a wide range of systems, some unsymmetric (Rozlozník and Simoncini 2002) or even indefinite (Meurant 2006). The key ingredient to convergence being the same as in the (symmetric) positive-definite case: the system matrix must exhibit clustered eigenvalues along with good conditioning.

Considering our discussion of K_s, K_x and K_y when we studied the spectral condition of the MINRES system, we can also expect a very good conditioning of the CG preconditioned system along with a large number of eigenvalues clustered in the vicinity of 1. But the skew-symmetric part of the matrix may clearly perturb the convergence of the algorithm in the long term.

Imperfect Recall Extension

There is a strong incentive to reduce the size of the game abstraction tree and allow the players to forget about the past (Waugh et al. 2009). However it is worth noting that (Koller and Megiddo 1992) proved that such games are NP-hard, hence out of reach of IPMs.

At some costs, we can force a game with imperfect recall to fit in the sequence form framework: one way is to allow the information nodes to collapse and merge at given moments. Sequence form seems to easily adapt to such extension when the forgotten information is anything but the action history. But there is an obvious catch: while realization weights can still be interpreted as strategy probabilities, they cease to be well defined. Indeed, the sequence form no longer represents — at less accurately — the move transitions. Consequently, solutions of the min-max equivalent problem are no longer genuine equilibria.

Rather than discarding this approach, we decide to regard it as an another approximation on top of the game abstraction. Hence we expect the solution to exhibit an additional *exploitability*, and possibly expect this exploitability to be compensated by a larger or more accurate abstraction. The interesting problem of quantifying and possibly bounding this exploitability is beyond the scope of this paper.

On the interior point side, there is one notable difference in the constraint structure, but without any consequences. Information sets will now originate from more than one predeccessor. Since a child node can now originate from all parent nodes with similar action history, the unique negative coefficient in each row of E and F is replaced by a set of negative coefficients (whose sum is -1). But since proposition 1 is still valid, the matrices E and F can be split similarly to the perfect recall case. Global convergence of both Krylov methods and interior point method are unaffected.

All the problems of the numerical results in the final section will make use of this imperfect recall extension.

Implementation

The implementation of the primal-dual path-following infeasible method is a very well documented subject. See (Wright 1997) in particular, for standard frameworks. The subjects like starting point, path proximity, path neighborhood, centering are not very different from the quadratic case. However we would like to address two particular subjects.

Krylov Method Termination

It is worth mentioning the termination condition to the Krylov method. Usually, iterative methods are stopped when a target accuracy is reached or when the method fails to converge. But the exact value of this target accuracy is difficult to decide beforehand.

In exact precision, Krylov methods would yield to an exact solution in at most the size of the system. In practice, when appropriately preconditioned, they converge *much* faster. In the numerical results section, MINRES needed less than 2% of this amount to obtain an acceptable solution, and the conjugate gradient needed less than 0.2%.

Indeed, it is widely known that the IPM does not require an accurate solution of the Newton equations to converge. This variation is known as the *inexact* Newton method. See for example (Monteiro and ONeal 2003). In practice, when an iterative method is used to solve the Newton equations, the required accuracy is merely the accuracy such that the approximate directions are accurate enough for a new effective IPM step.

The resulting technique is therefore easy to derive. The Krylov method is iterated without specific target accuracy. Regularly, the interior point method interrupts the Krylov solver and runs the line-search method with the current solver iterate. The line-search checks if any progress is possible using these search directions, and if any, computes the corresponding step length. The Krylov method is finally terminated when two conditions are verified: the search directions converged, and the step length converged (to a strictly positive value). This technique gives an easy practical condition to the Krylov termination problem and ensures that no unnecessary work is performed.

Scalability

The author has successfully run the method described in this paper to games up to 1.3 10^9 game states using a single workstation. The EGT and CFR methods are said to have successfully solved games with 10^{12} game states. In order to reach such orders of magnitudes, an algorithm must display a very high level of *scalability* both in term of memory and computing resources. This is key to exploit massively parallel architectures such as supercomputers.

As already noticed (Gondzio 2012b), for truly large problems, nothing prevents the IPMs to run in *matrix-free* regime provided that the newton equations are solved using iterative methods. Naturally, this supposes that the underlying problem is able to propose an alternative representation of the payoff and constraint matrices. When dealing with games like the Texas Hold'em poker, the matrices display some special characteristics that make them particularly easy to condense in smaller representations (Sandholm 2010; Gilpin et al. 2007). Succinctly, since the cards information can be separated to the action tree, the constraint and payoff matrices can be decomposed in Kronecker products of smaller matrices.

On the computational side, and preconditioning left aside, the Krylov solvers are essentially dominated by the matrix-vector multiplication. The matrices being either the payoff matrix or the constraint matrices. The preconditioner requires essentially the same operations, with the notable addition of the triangular solve operation of E_1 and F_1. The multiplication parallelization is easy and common in scientific supercomputing, but the triangular solve operation can be much harder due to its inherent sequential nature. However, we are reasonably optimistic that the special structure of the constraints could be again exploited in this situation.

Numerical Results

The following games (M1, M2, M3 and M4) are increasingly accurate, and therefore large, abstractions of the Texas Hold'em limit poker with the imperfect recall extension.

	Game states	A rows	A cols	A non-zero
M1	44,688,967	263,316	263,485	16,087,743
M2	86,150,842	488,541	488,710	37,962,365
M3	342,718,342	975,391	975,560	149,534,483
M4	770,330,842	1,462,241	1,462,410	333,230,233

The computations are carried out on a 12-core Xeon mid-range workstation with two dedicated GPU (Nvidia GeForce Titan, 6GB memory and 2880 cores). The interior point solver only utilizes one CPU core, but dispatches the computations to one or both GPUs concurrently. For memory considerations, M1, M2 and M3 are computed using one GPU. M4 utilizes two GPUs. The accuracy unit is the milli big blind per hand (mbb/h).

Our first observation on figure 1 is that CG clearly outperforms MINRES. MINRES is a very solid method, totally failsafe, with a steadily decreasing residual. The CG residual decreases less steadily, needs more monitoring, but converges considerably faster.

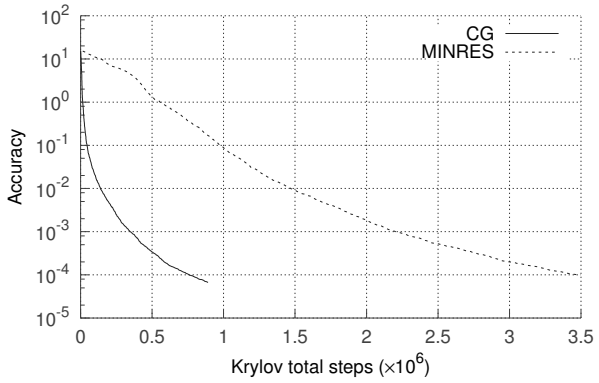

Figure 1: CG versus MINRES on M1

In the table below, the IPM uses CG to solve the abstractions down to 1 mbb/h. The column FLOP is the total number of double-precision floating-point operations considering only matrix-vector multiplications involving A or A^T. This serves as an estimate of the computational workload needed to solve the abstraction.

	Game states	Time	IPM steps	Krylov steps	FLOP $\times 10^{12}$
M1	44,688,967	9h59	286	349,797	22.7
M2	86,150,842	12h49	316	315,794	48.0
M3	347,718,342	46h05	443	274,113	165.6
M4	770,330,842	58h16	476	253,294	337.6

IPM convergence using CG displays solid results. Figure 2 suggests overall *linear* dependence over the game size (number of game states). This matches the standard practical complexity results of IPM (Wright 1997; Gondzio 2012a).

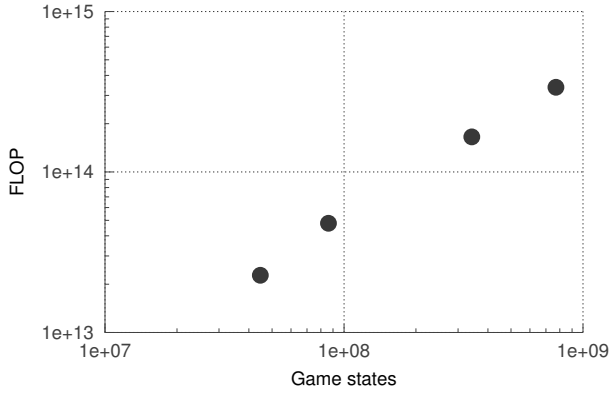

Figure 2: Dependence on the game size (M1 to M4)

IPMs have practical and theoretical dependence of $\mathcal{O}(\ln(1/\epsilon))$ on the required termination accuracy. This clearly does not hold in our IPM variation since the Newton system has to be parameterized with a target accuracy and the condition of the preconditioned system still slowly degrades. Consequently, despite the regularization, the steps of the IPM require more and more work and slowly decrease in efficiency. Nevertheless, figure 3 displays a solid convergence even past the target accuracy of 1 mbb/h.

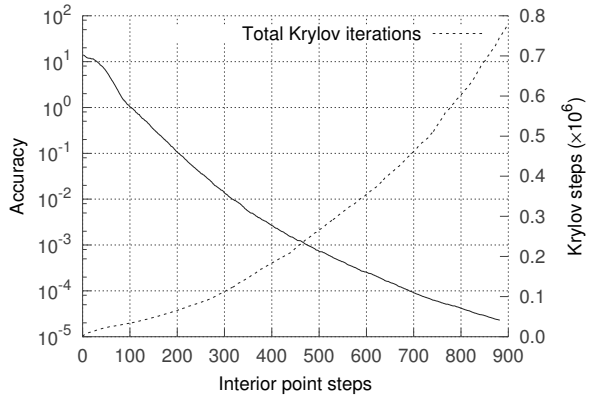

Figure 3: Dependence on accuracy on M1

Conclusion

We established that interior point methods can indeed solve very large instances of zero-sum games of incomplete information. We showed how projected Krylov methods, along with implicit factorization preconditioners, apply to such problems and have memory requirements linear in the size of the game.

We examined the spectral condition of two possible preconditioned systems and discussed the convergence of their corresponding Krylov methods. We demonstrated that the projected MINRES is theoretically applicable in the first case, and observed the practical efficiency of the projected conjugate gradient in the other.

Finally, we introduced an extension of the sequence form representation for approximating games of imperfect recall. We presented numerical results confirming the practical efficiency of the whole approach.

References

Benzi, M.; Golub, G. H.; and Liesen, J. 2005. Numerical solution of saddle point problems. *Acta numerica* 14:1–137.

Boyd, S. P., and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Dollar, H., and Wathen, A. 2004. Incomplete factorization constraint preconditioners for saddle point problems. Technical report, Technical Report 04/01, Oxford University Computing Laboratory, Oxford, England.

Faber, V., and Manteuffel, T. 1984. Necessary and sufficient conditions for the existence of a conjugate gradient method. *SIAM Journal on Numerical Analysis* 21(2):352–362.

Friedlander, M. P., and Orban, D. 2012. A primal–dual regularized interior-point method for convex quadratic programs. *Mathematical Programming Computation* 4(1):71–107.

Ghosh, A., and Boyd, S. P. 2003. Minimax and convex-concave games. University lecture notes available at http://www.stanford.edu/class/ee392o/cvxccv.pdf.

Gilpin, A.; Hoda, S.; Pena, J.; and Sandholm, T. 2007. Gradient-based algorithms for finding nash equilibria in extensive form games. In *Internet and Network Economics*. Springer. 57–69.

Gondzio, J. 2012a. Interior point methods 25 years later. *European Journal of Operational Research* 218(3):587–601.

Gondzio, J. 2012b. Matrix-free interior point method. *Computational Optimization and Applications* 51(2):457–480.

Gould, N.; Orban, D.; and Rees, T. 2013. Projected krylov methods for saddle-point systems. *Cahiers du GERAD* 23.

Hoda, S.; Gilpin, A.; and Pena, J. 2007. A gradient-based approach for computing nash equilibria of large sequential games. Unpublished manuscript available at http://andrew. cmu. edu/user/shoda/1719. pdf.

Keller, C.; Gould, N. I.; and Wathen, A. J. 2000. Constraint preconditioning for indefinite linear systems. *SIAM Journal on Matrix Analysis and Applications* 21(4):1300–1317.

Koller, D., and Megiddo, N. 1992. The complexity of two-person zero-sum games in extensive form. *Games and economic behavior* 4(4):528–552.

Koller, D.; Megiddo, N.; and Von Stengel, B. 1994. Fast algorithms for finding randomized strategies in game trees. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, 750–759. ACM.

Koller, D.; Megiddo, N.; and Von Stengel, B. 1996. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior* 14(2):247–259.

Meurant, G. 2006. *The Lanczos and conjugate gradient algorithms: from theory to finite precision computations*, volume 19. SIAM.

Monteiro, R. D., and ONeal, J. W. 2003. Convergence analysis of a long-step primal-dual infeasible interior-point lp algorithm based on iterative linear solvers. *Georgia Institute of Technology*.

Paige, C. C., and Saunders, M. A. 1975. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis* 12(4):617–629.

Romanovskii, I. 1962. Reduction of a game with complete memory to a matrix game. *Soviet Mathematics* 3:678–681.

Rozložník, M., and Simoncini, V. 2002. Krylov subspace methods for saddle point problems with indefinite preconditioning. *SIAM journal on matrix analysis and applications* 24(2):368–391.

Rubin, J., and Watson, I. 2011. Computer poker: A review. *Artificial Intelligence* 175(5):958–987.

Sandholm, T. 2010. The state of solving large incomplete-information games, and application to poker. *AI Magazine* 31(4):13–32.

Waugh, K.; Zinkevich, M.; Johanson, M.; Kan, M.; Schnizlein, D.; and Bowling, M. H. 2009. A practical use of imperfect recall. In *SARA*.

Wright, S. J. 1997. *Primal-dual interior-point methods*, volume 54. Siam.

Zinkevich, M.; Johanson, M.; Bowling, M. H.; and Piccione, C. 2007. Regret minimization in games with incomplete information. In *NIPS*.