A simple algorithm for computing Nash-equilibria in incomplete information games

Elvis Dohmatob
elvis.dohmatob@inria.fr
Parietal Team, INRIA / CEA, Neurospin, Université Paris-Saclay, France

Abstract

We present a simple projection-free primal-dual algorithm for computing approximate Nash-equilibria in two-person zero-sum sequential games with incomplete information and perfect recall (like Texas Hold'em Poker). Our algorithm is numerically stable, performs only basic iterations (i.e matvec multiplications, clipping, etc., and no calls to external first-order oracles, no matrix inversions, etc.), and is applicable to a broad class of two-person zero-sum games including simultaneous games and sequential games with incomplete information and perfect recall. The applicability to the latter kind of games is thanks to the sequence-form representation which allows one to encode such a game as a matrix game with convex polytopial strategy profiles. We prove that the number of iterations needed to produce a Nash-equilibrium with a given precision is inversely proportional to the precision. We present experimental results on matrix games on simplexes and Kuhn Poker.

1 Introduction

The sequence-form representation for two-person zero-sum games with incomplete information was introduced in [8], and the theory was further developed in [9, 17, 18] where it was established that for such games, there exist sparse matrices $A \in \mathbb{R}^{n_1 \times n_2}$, $E_1 \in \mathbb{R}^{l_1 \times n_1}$, $E_2 \in \mathbb{R}^{l_2 \times n_2}$, and vectors $e_1 \in \mathbb{R}^{l_1}$, $e_2 \in \mathbb{R}^{l_2}$ such that n_1, n_2, l_1, and l_2 are all linear in the size of the game tree (number of states in the game) and such that Nash-equilibria correspond to pairs (x, y) of *realization plans* which solve the primal LCP (Linear Convex Program)

$$\underset{(y,p) \in \mathbb{R}^{n_2} \times \mathbb{R}^{l_1}}{\text{minimize}}\; e_1^T p, \text{ subject to: } y \geq 0, E_2 y = e_2, -Ay + E_1^T p \geq 0, \tag{1}$$

and the dual LCP

$$\underset{(x,q) \in \mathbb{R}^{n_1} \times \mathbb{R}^{l_2}}{\text{maximize}}\; -e_2^T q, \text{ subject to: } x \geq 0, E_1 x = e_1, A^T x + E_2^T q \geq 0. \tag{2}$$

The vectors $p = (p_0, p_1, \ldots, p_{l_2 - 1}) \in \mathbb{R}^{l_2}$ and $q = (q_0, q_1, \ldots, q_{l_1 - 1}) \in \mathbb{R}^{l_1}$ are dual variables. A is the *payoff matrix* and each E_k is a matrix whose entries are -1, 0 or 1, with exactly 1 entry per row which equals -1 except for the first whose whose first entry is 1 and all the others are 0. Each of the vectors e_k is of the form $(1, 0, \ldots, 0) \in \mathbb{R}^{l_k}$.

The LCPs above have the equivalent saddle-point formulation

$$\underset{y \in Q_2}{\text{minimize}}\; \underset{x \in Q_1}{\text{maximize}}\; x^T A y, \tag{3}$$

where the compact convex polytope $Q_k := \{z \in \mathbb{R}^{n_k} | z \geq 0, E_k z = e_k\} \subseteq [0,1]^{n_k}$ is identified with the strategy profile of player k in the sequence-form representation [8, 9, 17, 18]. As a special case, the above formulation matrix games with complete information in which the strategy profiles are simplexes. Now, at a feasible point (y, p, x, q) for the LCPs, the duality gap $\tilde{G}(y, p, x, q)$ is given by

$$\tilde{G}(y, p, x, q) := e_1^T p - (-e_2^T q) = p_0 + q_0 = \max\{u^T A y - x^T A v | (u, v) \in Q_1 \times Q_2\} =: G(x, y) \geq 0. \tag{4}$$

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

In (4), the quantity $G(x, y)$ is nothing but the primal-dual gap for the equivalent saddle-point problem (3). It was shown (see Theorem 3.14 of [18]) that a pair $(x, y) \in Q_1 \times Q_2$ of realization plans is a solution to the LCPs (1) and (2) (i.e is a Nash-equilibrium for the game) if and only if there exist vectors p and q such that

$$-Ay + E_1^T p \geq 0,\ A^T x + E_2^T q \geq 0,\ x^T(-Ay + E_1^T p) = 0,\ y^T(A^T x + E_2^T q) = 0. \quad (5)$$

Moreover, at equilibria strong duality holds and $\tilde{G}(y, p, x, q) = G(x, y) = 0$.

Definition 1 (**Nash ϵ-equilibria**). *Given $\epsilon > 0$, a Nash ϵ-equilibrium is a pair (x^*, y^*) of realization plans for which there exists dual vectors p^* and q^* for problems* (1) *and* (2) *such that the duality gap at (y^*, p^*, x^*, q^*) does not exceed ϵ. That is, $0 \leq \tilde{G}(y^*, p^*, x^*, q^*) \leq \epsilon$.*

Quick sketch of our contribution. It should be noted that the class of games considered here (sequential games with incomplete information), the LCPs (1) and (2) are exceedingly larger than what state-of-the-art LCP and interior-point solvers can handle (see [7, 4]). Developing on an alternative notion of approximate equilibria (see Definition 2) homologous to that presented in Definition 1, we device in section 3 a simple numerically stable primal-dual algorithm Alg. 1 for computing approximate Nash-equilibria in sequential two-person zero-sum games with incomplete information and perfect recall. On each iteration, the only operations performed by our algorithm are of the form $A^T x$, Ay, $E_1^T p$, $E_2^T q$, and $(x)_+ := (\max(0, x_j))_j$. We also prove (Theorem 1) that –in an ergodic / Cesàrio sense– the number of iterations required by the algorithm to produce an approximation equilibrium to a precision ϵ is $\mathcal{O}(1/\epsilon)$, with explicit values for the constants involved.

2 Related work

In [7], a nested iterative procedure using the Excessive Gap Technique (EGT) [13] was used to solve the equilibrium problem (3). The authors reported a $\mathcal{O}(1/\epsilon)$ convergence rate (which derives from the general EGT theory) for the outer-most iteration loop. [4] proposed a modified version of the techniques in [7] and proved a $\mathcal{O}\left((\|A\|/\delta)\log(1/\epsilon)\right)$ convergence rate in terms of the number of calls made to a first-order oracle. Here $\delta = \delta(A, E_1, E_2, e_1, e_2) > 0$ is a certain condition number for the game. The crux of their technique was to observe that (3) can further be written a the minimization of the duality gap function $G(x, y)$ (defined in (4)) for the game[1], viz

$$\text{minimize}\{G(x, y) | (x, y) \in Q_1 \times Q_2\}, \quad (6)$$

and then show there exists a scalar $\delta > 0$ such that for any pair of realization plans $(x, y) \in Q_1 \times Q_2$,

$$\text{“distance between } (x, y) \text{ and set of equilibria”} \leq G(x, y)/\delta.$$

However, there are a number of issues, most notably: *(a)* The constant $\delta > 0$ can be arbitrarily small, and so the factor $\|A\|/\delta$ in the $\mathcal{O}\left((\|A\|/\delta)\log(1/\epsilon)\right)$ convergence rate can be arbitrarily large for ill-conditioned games. *(b)* The reported linear convergence rate is not in terms of basic operations (addition, multiplication, matvec, clipping, etc.), but in terms of the number of calls to a first-order oracle. Most notably, projections onto the polytopes Q_k are computed on each iteration.

Recently, [10] proposed accelerations to first-order methods for computing Nash-equilibria (including those just discussed), by an appropriate choice of the underlying *Bregman distance* and the *distance generating function* (essential ingredients in EGT-type algorithms). These modifications provably gain a constant factor in the worst-case convergence rate over the original algorithm.

3 Generalized Saddle-point Problem (GSP) for Nash-equilibrium

In the next theorem, we show that the LCPs (1) and (2) can be conveniently written as a GSP in the sense of [6]. The crux of idea is to “dualize” the linear constraints in the definitions of the strategy polytopes Q_k, by augmenting the payoff matrix to yield an equivalent saddle-point problem, involving only very simple functions like ReLUs and matvecs. We elaborate this construction in the following theorem, whose proof is provided in Appendix A.

[1] The minimizers of G are precisely the equilibria of the game.

Theorem 1. *Define two proper closed convex functions*

$$g_1 : \mathbb{R}^{n_2} \times \mathbb{R}^{l_1} \to (-\infty, +\infty], \quad g_1(y,p) := i_{y \geq 0} + e_1^T p$$
$$g_2 : \mathbb{R}^{n_1} \times \mathbb{R}^{l_2} \to (-\infty, +\infty], \quad g_2(x,q) := i_{x \geq 0} + e_2^T q$$

Also define two bilinear forms Ψ_1, $\Psi_2 : \mathbb{R}^{n_2} \times \mathbb{R}^{l_1} \times \mathbb{R}^{n_1} \times \mathbb{R}^{l_2} \to \mathbb{R}$ *by letting*

$$K := \begin{bmatrix} A & -E_1^T \\ E_2 & 0 \end{bmatrix}, \; \Psi_1(y,p,x,q) := \begin{bmatrix} x \\ q \end{bmatrix}^T K \begin{bmatrix} y \\ p \end{bmatrix},$$

with $\Psi_2 = -\Psi_1$, *and define the functions* $\hat{\Psi}_1$, $\hat{\Psi}_2 : \mathbb{R}^{n_2} \times \mathbb{R}^{l_1} \times \mathbb{R}^{n_1} \times \mathbb{R}^{l_2} \to (-\infty, +\infty]$ *by*

$$\hat{\Psi}_1(y,p,x,q) := \Psi_1(y,p,x,q) + g_1(y,p) \text{ if } y \geq 0, \; \hat{\Psi}_1(y,p,x,q) = \infty \text{ otherwise},$$
$$\hat{\Psi}_2(y,p,x,q) := \Psi_2(y,p,x,q) + g_2(x,q) \text{ if } x \geq 0, \; \hat{\Psi}_2(y,p,x,q) = \infty \text{ otherwise.}$$

Finally, define the sets $S_1 := \mathbb{R}_+^{n_2} \times \mathbb{R}^{l_1}$ *and* $S_2 := \mathbb{R}_+^{n_1} \times \mathbb{R}^{l_2}$, *and consider the GSP(*Ψ_1, Ψ_2, g_1, g_2*): Find a quadruplet* $(y^*, p^*, x^*, q^*) \in S_1 \times S_2$ *s.t* $\forall (y,p,x,q) \in S_1 \times S_2$, *we have*

$$\hat{\Psi}_1(y^*,p^*,x^*,q^*) \leq \hat{\Psi}_1(y,p,x^*,q^*), \text{ and } \hat{\Psi}_2(y^*,p^*,x^*,q^*) \leq \hat{\Psi}_2(y^*,p^*,x,q). \tag{7}$$

Then GSP(Ψ_1, Ψ_2, g_1, g_2*) is equivalent to the LCPs* (1) *and* (2), *i.e a quadruplet* $(y^*, p^*, x^*, q^*) \in \mathbb{R}^{n_2} \times \mathbb{R}^{l_1} \times \mathbb{R}^{n_1} \times \mathbb{R}^{l_2}$ *solves the LCPs* (1) *and* (2) *iff it solves GSP(*Ψ_1, Ψ_2, g_1, g_2*).*

4 The Algorithm

By Theorem 1, solving for a Nash-equilibrium for the game is equivalent to solving the GSP (7), which as it turns out, is simpler conceptually: e.g, we no longer need to compute the complicated orthogonal projections proj_{Q_k}.

Definition 2 (**Nash** (ϵ_1, ϵ_2)**-equilibria** [6])**.** *Given tolerance levels* $\epsilon_1, \epsilon_2 > 0$, *a Nash* (ϵ_1, ϵ_2)*-equilibrium for the GSP* (7) *is any quadruplet* (y^*, p^*, x^*, q^*) *for which there exists a perturbation vector* v^* *such that* $\|v^*\| \leq \epsilon_1$ *and* $v^* \in \partial_{\epsilon_2}[\hat{\Psi}_1(.,.,x^*,q^*) + \hat{\Psi}_2(y^*,p^*,.,.)](y^*,p^*,x^*,q^*)$, *where* $\partial_\epsilon(f) : x \mapsto \{v \in \mathcal{H} | f(z) \geq f(x) + v^T(z-x) - \epsilon, \forall z \in \mathcal{H}\}$ *denotes the* ϵ*-enlarged subdifferential of a function* f. *Such a vector* v^* *above is called a Nash* (ϵ_1, ϵ_2)*-residual at the point* (y^*, p^*, x^*, q^*).

The above definition is a generalization of the notion of Nash-equilibrium since: *(a)* exact Nash-equilibria correspond to Nash $(0,0)$-equilibria, and *(b)* Nash ϵ-equilibria (in the sense of Definition 1) correspond to Nash $(0,\epsilon)$-equilibria. Putting things together and using results from [6], we obtiain Alg. 1 below which solves (7), and therefore the original LCPs (1) and (2).

Algorithm 1 Primal-dual algorithm for computing Nash $(\epsilon, 0)$-equilibria for (7)

Require: $\epsilon > 0$; $(y^{(0)}, p^{(0)}, x^{(0)}, q^{(0)}) \in \mathbb{R}^{n_2} \times \mathbb{R}^{l_1} \times \mathbb{R}^{n_1} \times \mathbb{R}^{l_2}$.
Ensure: A Nash $(\epsilon, 0)$-equilibrium $(y^*, p^*, x^*, q^*) \in S_1 \times S_2$ for the GSP (7), with precision $\epsilon > 0$.
Initialize: $v^{(0)} \leftarrow 0$, $k \leftarrow 0$, $\lambda \leftarrow 1/\|K\|$ (See Appendix B.)
while $k = 0$ or $\frac{1}{k\lambda}\|v^{(k)}\| \geq \epsilon$ **do**
  $y^{(k+1)} \leftarrow (y^{(k)} - \lambda(A^T x^{(k)} + E_2^T q^{(k)}))_+, \;\; p^{(k+1)} \leftarrow p^{(k)} - \lambda(e_1 - E_1 x^{(k)})$
  $x^{(k+1)} \leftarrow (x^{(k)} + \lambda(A y^{(k+1)} - E_1^T p^{(k+1)}))_+, \;\; \Delta x^{(k+1)} \leftarrow x^{(k+1)} - x^{(k)}$
  $\Delta q^{(k+1)} \leftarrow \lambda(E_2 y - e_2), \;\; q^{(k+1)} \leftarrow q^{(k)} + \Delta q^{(k+1)}$
  $y^{(k+1)} \leftarrow y^{(k+1)} - \lambda(A^T \Delta x^{(k+1)} + E_2^T \Delta q^{(k+1)}), \;\; \Delta y^{(k+1)} \leftarrow y^{(k+1)} - y^{(k)}$
  $p^{(k+1)} \leftarrow p^{(k+1)} + \lambda E_1 \Delta x^{(k+1)}, \;\; \Delta p^{(k+1)} \leftarrow p^{(k+1)} - p^{(k)}$
  $v^{(k+1)} \leftarrow v^{(k)} + (\Delta y^{(k+1)}, \Delta p^{(k+1)}, \Delta x^{(k+1)}, \Delta q^{(k+1)})$
  $k \leftarrow k + 1$
end while

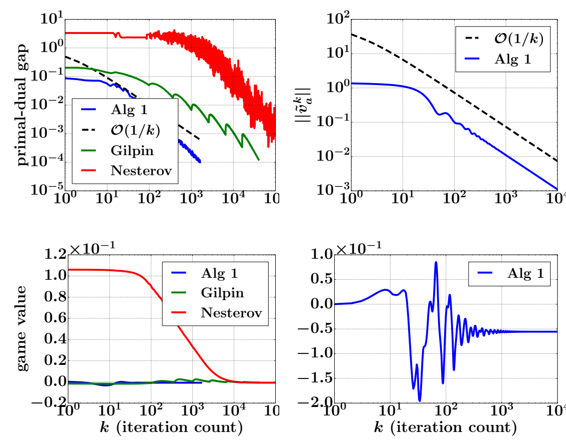
Theorem 2 (Ergodic / Cesàrio $\mathcal{O}(1/\epsilon)$ convergence of Alg. 1)**.** *Let* d_0 *be the euclidean distance between the starting point* $(y^{(0)}, p^{(0)}, x^{(0)}, q^{(0)})$ *of Algorithm 1 and the set of equilibria for the GSP* (7). *Then given any* $\epsilon > 0$, *there exists an index* $k_0 \leq \frac{2d_0\|K\|}{\epsilon}$ *such that after* k_0 *iterations the algorithm produces a quadruplet* $(y^{k_0}, p^{k_0}, x^{k_0}, q^{k_0})$ *and a vector* v^{k_0} *such that* $\|v_a^{k_0}\| \leq \epsilon$ *and* $v_a^{k_0} \in \partial[\hat{\Psi}_1(.,.,x^{k_0},q^{k_0}) + \hat{\Psi}_2(y^{k_0},p^{k_0},.,.)](y^{k_0},p^{k_0},x^{k_0},q^{k_0})$, *where* $v_a^{(k_0)} := \frac{1}{k\lambda}v^{(k_0)}$. *Thus Algorithm 1 outputs a Nash* $(\epsilon, 0)$*-equilibrium for the GSP* (7) *in at most* $\frac{2d_0\|K\|}{\epsilon}$ *iterations.*

5 Numerical experiments and results

Experiments. *(a)* Basic test-bed: Matrix games on simplexes. As in [14, 3], we generate a $10^3 \times 10^3$ random matrix whose entries are uniformly identically distributed in the closed interval $[-1, 1]$.

(b) Kuhn Poker, a "toy" incomplete-information sequential game. This game is a simplified form of Poker developed by Harold W. Kuhn in [11]. It already contains all the complexities (sequentiality, imperfection of information, etc.) of a full-blown Poker game like Texas Hold'em, but is simple enough to serve as a proof-of-concept for the ideas developed in this manuscript. The deck includes only three playing cards: a King, Queen, and Jack. One card is dealt to each player, then the first player must bet or pass, then the second player may bet or pass. If any player chooses to bet the opposing player must bet as well ("call") in order to stay in the round. After both players pass or bet, the player with the highest card wins the pot.

Results. The results of the experiments are shown in Fig. 1. We stress that the algorithms of Nesterov [14] and Gilpin [4] are included in the plots only indicatively, since this is not meant to be a benchmark. These reference algorithms are easy to have been included where not profiled in the "Kuhn poker" experiment because they are rather difficult to implement mainly due to the the need to project onto the strategy polytopes Q_k. One can see the linear (i.e exponentially fast) behavior of the algorithm in [4], in-between consecutive breakpoints on the ϵ grid (though the rate of linear convergence seems to by quite close to 1 here). As expected, the first-order smoothing algorithm labeled "Nesterov" [14] jitters around as the iterations go on because even the smoothed problem becomes heavily ill-conditioned near solutions. On the other hand, for both experiments, our proposed algorithm Alg. 1 behaves empirically as its proven $\mathcal{O}(1/\epsilon)$ convergence rate.



(a) matrix game on simplex (b) Kuhn poker

Figure 1: Convergence curves of Algorithm 1. In the top plots, we show the modified duality gap defined in Theorem 2. In the top plots, we show the evolution of the value ${x^{(k)}}^T A y^{(k)}$ of the games across iterations.

6 Concluding remarks

Making use of the sequence-form representation [8, 17, 18], we have proposed an equivalent Generalized Saddle-Point (GSP) formulation to the Nash-equilibrium problem for sequential games with incomplete information and perfect recall (e.g Texas Hold'em, etc.). Then, we have devised a simple numerically stable primal-dual algorithm for solving the GSP, and so by equivalence, compute (approximate) Nash-equilibria for such games. Our algorithm is simple to implement, with a low constant cost per iteration, and enjoys a rigorous convergence theory with a proven $\mathcal{O}(1/\epsilon)$ convergence in terms of basic operations (matvec products, clipping, etc.), to a Nash $(\epsilon, 0)$-equilibrium of the game. On an experimental footing, this is just preliminary works, and in future we plan to run more experiments on real Poker games to measure the practical power of the proposed algorithm compared to state-of-the-art methods like CFR and variants [20, 12, 1, 19], and first-order methods like [4].

Future work. One of the NIPS OPT2016 reviewers kindly pointed to our attention that the $\mathcal{O}(1/\epsilon)$ convergence rate for Alg. 1 established in Theorem 1 could be made linear (i.e exponentially fast). We plan to do some work in this direction.

Acknowledgment. This work has been funded by EU FP7/2007-2013 under grant agreement no. 604102, Human Brain Project (HBP) and the iConnectome Digiteo.

References

[1] M. Bowling et al. “Heads-up limit hold’em poker is solved”. In: *Science* 347 (2015).

[2] N. Brown, S. Ganzfried, and T. Sandholm. “Hierarchical Abstraction, Distributed Equilibrium Computation, and Post-Processing, with Application to a Champion No-Limit Texas Hold’em Agent”. In: *AAMAS*. 2015.

[3] A. Chambolle and T. Pock. “On the ergodic convergence rates of a first-order primal–dual algorithm”. In: *Math. Program.* (2015).

[4] A. Gilpin, J. Peña, and T. Sandholm. “First-order algorithm with $\mathcal{O}(\ln(1/\epsilon))$ convergence for ϵ-equilibrium in two-person zero-sum games”. English. In: *Math. Program.* 133 (2012).

[5] A. Gilpin and T. Sandholm. “Lossless Abstraction of Imperfect Information Games”. In: *J. ACM* 54 (2007).

[6] Y. He and R. D. Monteiro. “Accelerating block-decomposition first-order methods for solving generalized saddle-point and Nash equilibrium problems”. In: *Optimization-online preprint* (2013).

[7] S. Hoda et al. “Smoothing techniques for computing Nash equilibria of sequential games”. In: *Math. Oper. Res.* 35 (2010).

[8] D. Koller and N. Megiddo. “The complexity of two-person zero-sum games in extensive form”. In: *GEB* 4 (1992).

[9] D. Koller, N. Megiddo, and B. Von Stengel. “Fast algorithms for finding randomized strategies in game trees”. In: *STOC*. ACM. 1994.

[10] C. Kroer et al. “Faster First-Order Methods for Extensive-Form Game Solving”. In: *EC’15*. 2015.

[11] H. W. Kuhn. “A simplified two-person poker”. In: *Annals of Math* (1950).

[12] M. Lanctot et al. “Monte Carlo sampling for regret minimization in extensive games”. In: *NIPS*. 2009.

[13] Y. Nesterov. “Excessive gap technique in nonsmooth convex minimization”. In: *SIAM J. Optim.* 16 (2005).

[14] Y. Nesterov. “Smooth minimization of non-smooth functions”. In: *Math. Program.* Ser. A 103 (2005).

[15] B. O’Donoghue and E. Candes. “Adaptive restart for accelerated gradient schemes”. In: *FOUND COMPUT MATH* 15 (2013).

[16] T. Sandholm. “Abstraction for solving large incomplete-information games”. In: *AAAI* (2015).

[17] B. von Stengel. “Efficient computation of behavior strategies”. In: *GEB* 14 (1996b).

[18] B. von Stengel. *Equilibrium Computation for Two-player Games in Strategic and Extensive Form. N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani (eds.), Algorithmic Game Theory*. Cambridge Univ. Press, Cambridge UK.

[19] O. Tammelin. “Solving Large Imperfect Information Games Using CFR+”. In: *CoRR* abs/1407.5042 (2014).

[20] M. Zinkevich et al. “Regret minimization in games with incomplete information”. In: *NIPS* (2008).

A Proofs

Proof of Theorem 1. It suffices to show that at any point $(y, p, x, q) \in S_1 \times S_2$, the duality gap between the primal LCP (1) and the dual LCP (2) equals the duality gap of GSP(Ψ_1, Ψ_2, g_1, g_2). Indeed, the unconstrained objective in (1), say $a(x, y)$, can be computed as

$$\begin{aligned}
a(y,p) &= e_1^T p + i_{y \geq 0} + i_{-Ay+E_1^T p \geq 0} + i_{E_2 y = e_2} \\
&= g_1(y,p) + \max_{x' \geq 0} x'^T(Ay - E_1^T p) + \max_{q'} q'^T(E_2 y - e_2) \\
&= g_1(y,p) + \max_{x',q'} x'^T Ay - x'^T E_1^T p + q'^T E_2 y (i_{x' \geq 0} + e_2^T q) \\
&= g_1(y,p) - \min_{x',q'} \Psi_2(y,p,x',q') + g_2(x',q') = g_1(y,p) - \underbrace{\min_{x',q'} \hat{\Psi}_2(y,p,x',q')}_{\phi_2(y,p)} \\
&= g_1(y,p) - \phi_2(y,p).
\end{aligned}$$

Similarly, the unconstrained objective, say $b(x, q)$, in the dual LCP (2) writes

$$\begin{aligned}
b(x,q) &= -q^T e_2 - i_{x \geq 0} - i_{A^T x + E_2^T q \geq 0} - i_{E_1 x = e_1} \\
&= -g_2(x,q) + \min_{y' \geq 0} y'^T(A^T x + E_2^T q) + \min_{p'} p'^T(e_1 - E_1 x) \\
&= -g_2(x,q) + \min_{y',p'} \Psi_1(y',p',x,q) + g_1(y',p') = -g_2(x,q) + \underbrace{\min_{y',p'} \hat{\Psi}_1(y',p',x,q)}_{\phi_1(x,q)} \\
&= -g_2(x,q) + \phi_1(x,q).
\end{aligned}$$

Thus, noting that $-\infty < \phi_1(x, q), \phi_2(y, p) < +\infty$ (so that all the operations below are valid), one computes the duality gap between the primal LCP (1) and dual the LCP (2) at (y, p, x, q) as

$$\begin{aligned}
a(y,p) - b(x,q) &= g_1(y,p) - \phi_2(y,p) + g_2(x,q) - \phi_1(x,q) \\
&= \Psi_1(y,p,x,q) + g_1(y,p) - \phi_2(y,p) + \Psi_2(y,p,x,q) + g_2(x,q) - \phi_1(x,q) \\
&= \hat{\Psi}_1(y,p,x,q) - \phi_1(x,q) + \hat{\Psi}_2(y,p,x,q) - \phi_2(y,p) \\
&= \text{duality gap of GSP}(\Psi_1, \Psi_2, g_1, g_2) \text{ at } (y,p,x,q),
\end{aligned}$$

where the second equality follows from $\Psi_1 + \Psi_2 := 0$. □

Proof of Theorem 2. It is clear that the quadruplet $(\Psi_1, \Psi_2, g_1, g_2)$ satisfies assumptions B.1, B.2, B.3, B.5, and B.6 of [6] with $L_{xx} = L_{yy} = 0$ and $L_{xy} = L_{yx} = \|K\|$. Now, one easily computes the proximal operator of g_j in closed-form as $\text{prox}_{\lambda g_j}(a, b) \equiv ((a)_+, b - \lambda e_j)$. With all these ingredients in place, Algorithm 1 is then obtained from Algorithm T-BD of [6] applied on the GSP (7) with the choice of parameters: $\sigma = 1 \in (0, 1]$, $\sigma_x = \sigma_y = 0 \in [0, \sigma)$, $\lambda_{xy} := \frac{1}{\sigma L_{xy}}\sqrt{(\sigma^2 - \sigma_x^2)(\sigma^2 - \sigma_y^2)} = \sigma/\|K\| = 1/\|K\|$, and $\lambda = \lambda_{xy} \in (0, \lambda_{xy}]$. The convergence result then follows immediately from Theorem 4.2 of [6]. □

B Practical considerations

Computing $\|K\|$. A major ingredient in the proposed algorithm is $\|K\|$, the 2-norm of the huge matrix K. This can be efficiently computed using the power iteration. Also since $\|K\|$ is only used in defining the step-size $\lambda := 1/\|K\|$, it may be possible to avoid computing $\|K\|$ altogether, and instead use a line-search / backtrack strategy (see [15], e.g) for setting λ.

Efficient computation of matvec products Ay and $A^T x$. In Alg. 1, most of the time is spent pre-multiplying vectors by A and A^T. For *flop-type* Poker games like Texas Hold'em and Rhode Island Hold'em, A (and thus A^T too) is very big (up 10^{14} rows and columns!) but is sparse and has a rich block-diagonal structure (each block is itself the Kronecker product of smaller matrices) which can be carefully exploited, as in [7]. Also the sampling strategies presented in the recent work [10] (section 6), for generating unbiased estimates of Ay and $A^T x$ would readily convert Algorithm 1 into an online and much scalable solver.

Game abstraction. For many variants of Poker, there has been extensive research in lossy / lossless abstraction techniques (for example [5] and more recently, [16, 2]), wherein strategically equivalent or not-so-different situations in the game tree are lumped together. This can drastically reduce the size of the state space from a player's perspective, and ultimately, the size of the matrices A, E_1, and E_2, without significantly deviating much from the true game.