Continuous Local Search

Constantinos Daskalakis
EECS and CSAIL, MIT[*]

Christos Papadimitriou
Computer Science, UC Berkeley[†]

Abstract

We introduce CLS, for continuous local search, a class of polynomial-time checkable total functions that lies at the intersection of PPAD and PLS, and captures a particularly benign kind of local optimization in which the domain is continuous, as opposed to combinatorial, and the functions involved are continuous. We show that this class contains several well known intriguing problems which were heretofore known to lie in the intersection of PLS and PPAD but were otherwise unclassifiable: Finding fixpoints of contraction maps, the linear complementarity problem for P matrices, finding a stationary point of a low-degree polynomial objective, the simple stochastic games of Shapley and Condon, and finding a mixed Nash equilibrium in congestion, implicit congestion, and network coordination games. The last four problems belong to CCLS, for convex CLS, another subclass of PPAD ∩ PLS seeking the componentwise local minimum of a componentwise convex function. It is open whether any or all of these problems are complete for the corresponding classes.

1 Introduction

The complexity class TFNP (search problems that always have a solution) contains some of the most fundamental, elegant, and intriguing computational problems such as factoring and Nash equilibrium. It is a "semantic" class (see [18]), in the sense there is no general way of telling whether an NP machine always has a witness, and hence it is unlikely to have complete problems. Hence, its various subclasses corresponding to "proof styles" of the existence theorem implicit in every problem in TFNP have been studied extensively and productively: PPP (pigeonhole principle, [19]), PLS (local search [11]), PPA (parity argument, [19]), PPAD (fixpoint problems, [19, 7, 6]) etc., and many were found to teem with interesting complete problems. Of these, the complexity class PLS captures *local search*, that is, the computation of a local optimum of a potential in a discrete solution space equipped

with an adjacency relation. This computational mode is important partly because often finding such a local optimum is not a compromise to global optimality, but the precise objective of the problem in hand. A problem in PLS is specified by two functions, one, N, mapping the solution space, say $\{0,1\}^n$, to itself (intuitively, providing for each possible solution a candidate better solution), and another, p, mapping the solution space to the integers (depending on your point of view, the potential function or the objective to be locally optimized). The solution sought is any solution $s \in \{0,1\}^n$ such that $p(N(s)) \geq p(s)$.[1] Many important computational problems belong to this class, and several have been shown PLS-complete (see, for example, [22]). In contrast, the class PPAD captures approximate fixpoints, and hence its typical problem can be naturally defined geometrically, in terms of a function f mapping, say, $[0,1]^3$ to itself (perhaps presented as an arithmetic circuit), and seeking an approximate fixpoint for some given accuracy; the function f needs to be continuous with a given Lipschitz constant, and so the sought output is either an approximate fixpoint or a violation of Lipschitz continuity (the latter option can be made moot by resorting to representations of f, such as arithmetic circuits with $\{\pm, *, \max, \min\}$ gates, in which Lipschitz continuity is guaranteed and self-evident).

It turns out (see Theorem 2.1) that any problem in PLS can also be embedded in geometry, by encoding the solution space into "cubelets" in the 3-dimensional unit cube, say. The result is two functions $f : [0,1]^3 \to [0,1]^3$ and $p : [0,1]^3 \to [0,1]$, where f is the neighborhood function and p the potential function, and p is assumed Lipschitz continuous, while f is not necessarily continuous. We are asked to find an approximate fixpoint of f *with respect to* p, that is, a point x such that $p(f(x)) \geq p(x) - \epsilon$, for some given $\epsilon > 0$, where the choice of ϵ is paired with the Lipschitz continuity of p

---

[1]More usually, the definition of PLS is in terms of a neighborhood correspondence $\mathcal{N}$ providing for every point in the solution space $\{0,1\}^n$ a list of candidate better solutions; in this case, we need to find any solution $s \in \{0,1\}^n$ that is not larger than any solution in $\mathcal{N}(s)$. It is easy to that the two definitions, i.e. in terms of a neighborhood map or a neighborhood correspondence, are equivalent.

to make sure that a solution of polynomial description length exists. In other words, local search can be seen as a variant of the search for a fixpoint, in which the existence of an approximate fixpoint is guaranteed not by the function's continuity, but by a potential argument.

Looking at PLS and PPAD this way, as close relatives so to speak, is particularly helpful when one considers the class PLS ∩ PPAD (which has been mentioned often recently because it contains many natural problems, see immediately below). Let us begin by pointing out that, unlike NP ∩ coNP which is a semantic class and thus unlikely to have complete problems, PPAD ∩ PLS does have a complete problem, implied by the discussion above, namely the problem EITHER FIXPOINT: Given three functions f, g and p, and $\epsilon, \lambda > 0$, find an approximate fixpoint of f (or violation of f's λ-continuity) *or* an approximate fixpoint of g with respect to p (or violation of p's λ-continuity); for the proof that this problem is complete for PPAD ∩ PLS see Corollary 2.3. However, there is something awkward about this characterization; namely, *we seek fixpoints of two unrelated functions.* Hence, the following is a most interesting open question about PPAD ∩ PLS: If in the above problem, f and g coincide in a single function that is both continuous and driven by a continuous potential, does the resulting problem still capture all of PPAD ∩ PLS? We conjecture that it does not, and this conjecture is the starting point for the definition of our new class.

Coming now from the other direction of motivation — proceeding from problems to classes instead of from larger to smaller classes — during the past few years researchers noticed that several intriguing computational problems related to fixpoints and games, but not only, lie within the intersection of PPAD and PLS.

- For example, finding a min-max strategy in the simple stochastic games of Shapley and Condon [23, 5] has been thus classified [8] (see [12] for an earlier proof).

- It was also noted [4] that finding a mixed Nash equilibrium for *congestion games* (in which players choose among subsets of resources, and incur delays depending on the resulting load on each resource) is a problem in the intersection of PLS and PPAD (essentially because the pure equilibrium problem is in PLS, and in fact PLS-complete [9], while finding a mixed Nash equilibrium for any game is a problem in PPAD [17, 19]).

- The same problem for *implicit congestion games*, in which the resources are edges on a network and the subsets that constitute the strategy set of each player are all paths between two endpoints.

- Ditto for the problem of finding a Nash equilibrium of a *network coordination game* (in which vertices of a graph play a game with each of their neighbors, adding the payoffs, where in each one of these two-person games and each pair of choices both players get the same payoff) is also in the intersection of PPAD and PLS [4].

- An intriguing and deceivingly simple problem in numerical optimization (which had not been considered before in the context of PLS and PPAD) generalizes the last three problems: Find any approximate stationary point of a given multivariate polynomial [16, 26]. Under certain conditions we show that it is also in the intersection of PLS and PPAD.

- Finding a solution to a linear complementarity problem with a P-matrix (see [14] for this elegant and important swarm of computational problems from linear optimization) also belongs to both PPAD (this had been known since [19]) and PLS [24, 15, 13].

- Finding an approximate fixpoint of a contraction map [1] (that is a continuous function f such that $|f(x) - f(y)| \leq c \cdot |x - y|$ for some $c < 1$) is an apparently easier special case of the (PPAD-complete) general problem, which, by the above reasoning, can be seen to lie in the intersection of PPAD and PLS, since $|f(x) - x|$ can serve as a potential function.

On the basis of multiple attempts for a polynomial-time solution resulting in very little progress (the usual circumstantial evidence in all of Complexity), it is tempting to conjecture that none of these problems can be solved in polynomial time. Are any of these problems complete for PPAD ∩ PLS? This is an important open question. *In this paper we identify a subclass of* PPAD ∩ PLS, which we call CLS for *continuous local search*, that contains all of the above problems. Thus completeness for PPAD ∩ PLS of any of these would imply a certain class collapse. The typical (and complete) problem in CLS is the following problem, which we call CONTINUOUS LOCALOPT. Given two functions f and p assumed to be Lipschitz continuous, and $\epsilon, \lambda > 0$, find an ϵ-approximate fixpoint of f with respect to p, or two points that violate the λ-continuity of p *or of* f. In other words, CLS is a close relative to PLS, in which not only the potential function is continuous, but also the neighborhood function is continuous. It is the continuity of the neighborhood function that also places it inside PPAD; indeed, an approximate fixpoint of the neighborhood function (which can be found within PPAD) is a solution.

We also identify another subclass of PPAD ∩ PLS, which we call CCLS for *convex* CLS, containing the first four problems in the above list. Call a function $f : \Re^n \mapsto \Re^n$ *componentwise convex* if it is a convex function of each of its variables when the remaining variables have been fixed (our precise definition given in Section 4 is a little more general in that it allows for a "component" to be a subset of the variables restricted to be in a convex domain such as a simplex). A problem in CCLS calls for a componentwise local minimum of such a function (or a violation of componentwise convexity). What is interesting about CCLS is that minimization can be achieved via a "distributed update" method, which however may not always decrease the objective — and hence it is not immediate that CCLS ⊆ CLS. We leave whether any of the seven problems are complete for the respective classes, as a challenging open question.

Finally, while PPAD seeks approximate fixpoints, Etessami and Yannakakis [8] have defined an apparently much larger class, FIXP, of problems seeking the binary representation of an actual exact fixpoint. The reason why we believe this class is larger is because several problems not known or believed to be in NP can be reduced to it, such as that of finding the bits of the sum of square roots of integers [10, 20, 25]. It would be interesting to see how these new smaller classes, CLS and CCLS, fare under the exactness restriction. In Section 5, we show that the sum-of-square-roots problem can be reduced to the exact version of CCLS and of CLS, and hence also the exact version of the continuous definition of PLS (Remarks 5.1 and 5.2).

2 Classes

We shall be considering problems in TFNP, that is, search problems with succinct and efficiently verifiable solutions that always have a solution. This is a surprisingly rich and diverse class, see [19]. Note that, in connection with total functions, a *reduction* from problem A to problem B is a polynomial-time function that maps an instance x of A to an instance $f(x)$ of B, plus another polynomial-time function that maps any solution y of $f(x)$ to a solution $g(y)$ of x. It is of interest to further classify problems in TFNP in terms of the proof technique establishing that the function is total. The class PPAD can be defined as the class of all such problems that can be reduced to the problem END OF THE LINE: Given n and two Boolean circuits S and P with n input bits and n output bits (intuitively, mapping each node in the solution space $\{0,1\}^n$ to a candidate predecessor and a candidate successor), such that 0^n has no predecessor and is a source (that is, $S(P(0^n)) \neq 0^n$) but does have a successor (that is, $P(S(0^n)) = 0^n$)), find another source or a sink. The class PLS [11] can be defined as

the class of all problems reducible to LOCALOPT: Given n and two Boolean circuits f and p with n inputs and outputs, where the output of p is interpreted as an integer, find an x such that $p(f(x)) \geq p(x)$. There are several other interesting subclasses of TFNP, such as the pigeonhole class PPP, the "sink only" version of PPAD called PPADS, and the undirected parity class PPA; see Figure 1 for a diagram of known inclusions; there are oracle results falsifying many of the remaining potential inclusions [3].
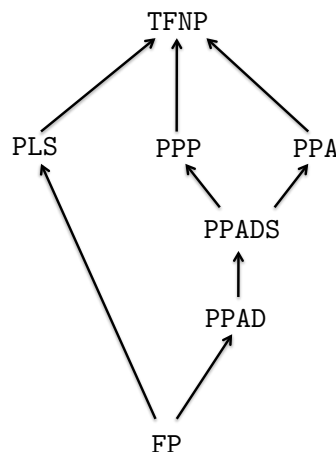

Figure 1: A snapshot of TFNP.

In this paper we focus on functions from continuous domains to continuous domains, and we shall represent these functions in terms of arithmetic circuits with operations $+, -, *,$ max, min, and $>$, the latter defined as $> (x, y) = 1$ if $x > y$ and 0 otherwise; rational constants are also allowed. If $>$ is not used in our circuits, the resulting functions are de facto continuous. The outputs of arithmetic circuits can be restricted in $[0, 1]$ by redefining the arithmetic gates to output 0 or 1 when the true output is negative or greater than one, respectively. We start by pointing out that both PPAD and PLS can be defined in terms of fixpoints of real-valued functions, represented by arithmetic circuits. PPAD is captured by the problem BROUWER FIXPOINT: Given an arithmetic circuit f with 3 real inputs and outputs, an $\epsilon > 0$ (the accuracy) and a $\lambda > 0$ (the purported Lipschitz constant of f in some p-norm), find (the succinct rational representation of) an approximate fixpoint of the function f, that is, a point $x \in \Re^3$ such that $|f(x) - x| < \epsilon$, or two points x, x' violating the λ-Lipschitz continuity of f, that is $|f(x) - f(x')| > \lambda |x - x'|$. We show next that PLS can captured by the problem REAL LOCALOPT: Given two arithmetic circuits computing two functions $f : [0, 1]^3 \mapsto [0, 1]^3$

and $p : [0,1]^3 \mapsto [0,1]$, an $\epsilon > 0$ (the accuracy) and a λ (the purported Lipschitz constant of p), find a point $x \in [0,1]^3$ such that $p(f(x)) \geq p(x) - \epsilon$, or two points x, x' violating the λ-Lipschitz continuity of p. Notice that this is indeed a total function: starting at an arbitrary point x, we can just follow the chain $x, f(x), f(f(x)), \ldots$ for $\frac{p(x)}{\epsilon}$ steps, as long as a step results in a more than ϵ decrease of the value of p.

THEOREM 2.1. REAL LOCALOPT *is* PLS-*complete.*

Proof. It is easy to see that any problem in PLS can be reduced to REAL LOCALOPT, by embedding the solution space in small cubelets of $[0,1]^3$. At the centers of the cubelets the values of f and p are defined so that they capture the neighborhood function and potential of the original problem. Then p is extended to the rest of the cube continuously by interpolation. f need not be continuous and is extended carefully so that no new solutions are introduced. More precisely, for a given instance (f, p) of LOCALOPT, our instance $(f', p', \lambda, \epsilon)$ of REAL LOCALOPT satisfies the property that, for all $x, y, z, y', z' \in [0,1]$, $p'(x, y, z) = p'(x, y', z')$ and $f'(x, y, z) = f'(x, y', z')$; in other words only the value of x is important in determining the values of p' and f'. Now, for every n-bit string s, if $x(s) \in \{0, \ldots, 2^n - 1\}$ is the number corresponding to s, we define $p'(x(s) \cdot 2^{-n}, y, z) = p(s)2^{-n}$, for all $y, z \in [0,1]$, and $f'(x(s) \cdot 2^{-n}, y, z) = (x(f(s)) \cdot 2^{-n}, y, z)$, for all $y, z \in [0,1]$. To extend f' and p' to the rest of the cube we do the following. p' is extended simply by interpolation, i.e. for $x = i2^{-n} + t \cdot (i + 1)2^{-n}$, where $t \in [0,1]$ and $i \in \{0, \ldots, 2^n - 2\}$, we define $p'(x, y, z) = p'(i2^{-n}, y, z) + tp'((i + 1)2^{-n}, y, z)$, for all $y, z \in [0,1]$. Exceptionally, for $x > 1 - 2^{-n}$, we define $p'(x, y, z) = p'(1 - 2^{-n}, y, z)$, for all y, z. We have to be a bit more careful in how we extend f' to the rest of the cube, so that we do not introduce spurious solutions. For all $i \in \{0, \ldots, 2^n - 2\}$, if $p'(i2^{-n}, y, z) < p'((i + 1)2^{-n}, y, z)$, we set $f'(i2^{-n} + t \cdot (i + 1)2^{-n}, y, z)$ to be equal to $f'(i2^{-n}, y, z)$, for all $t, y, z \in [0,1]$, while if $p'(i2^{-n}, y, z) \geq p'((i + 1)2^{-n}, y, z)$, we set $f'(i2^{-n} + t \cdot (i + 1)2^{-n}, y, z)$ to be equal to $f'((i + 1)2^{-n}, y, z)$, for all $t, y, z \in [0,1]$. Exceptionally, for $x > 1 - 2^{-n}$, we define $f'(x, y, z) = f'(1 - 2^{-n}, y, z)$, for all y, z. Finally, we choose $\epsilon = 0$, and $\lambda = 2^n$, so that p is guaranteed to be λ-Lipschitz continuous. It is easy to verify that any solution to the REAL LOCALOPT instance that we just created can be mapped to a solution of the PLS instance (f, p) that we departed from.

We point out next that REAL LOCALOPT is in PLS, by showing a reduction in the opposite direction, from real-valued to discrete. Suppose we are given an instance of REAL LOCALOPT defined by a four-tuple

$(f, g, \epsilon, \lambda)$. We describe how to reduce this instance to an instance (f', p') of LOCALOPT. For a choice of n that makes $2^{-n/3}$ sufficiently small with respect to ϵ and λ, we identify the n-bit strings that are inputs to f' and p' with the points of the 3-dimensional unit cube whose coordinates are integer multiples of $2^{-n/3}$. The value of p' on an n-bit string x is then defined to be equal to the value of p on x (viewed as a point in the cube). Similarly, the value of f' on an n-bit string x is defined by rounding each coordinate of $f(x)$ down to the closest multiple of $2^{-n/3}$. Suppose now that we have found a solution to LOCALOPT, that is an n-bit string x such that $p'(f'(x)) \geq p'(x)$. Notice that $f(x)$ and $f'(x)$ are within $2^{-n/3}$ to each other in the ℓ_∞ norm. Hence, it should be that $|p(f(x)) - p(f'(x))|_\infty < \lambda \cdot 2^{-n/3}$ (assuming that λ is the purported Lipschitz constant of p in the ℓ_∞ norm). If this is not the case, we have found a violation of the Lipschitzness of p. Otherwise, we obtain from the above that $p(f(x)) \geq p(x) - \lambda \cdot 2^{-n/3}$, where we used that, for an n-bit string x, $p'(x) = p(x)$ and $p'(f'(x)) = p(f'(x))$. If n is chosen large enough so that $\epsilon > \lambda \cdot 2^{-n/3}$, x is a solution to REAL LOCALOPT. In the above argument we assumed that λ is the Lipschitz constant of f in the ℓ_∞ norm, but this is not important for the argument to go through.

Finally, we consider the class PPAD $\cap$ PLS, and the problem EITHER FIXPOINT: Given three arithmetic circuits computing functions $f, g : [0,1]^3 \mapsto [0,1]^3$ and $p : [0,1]^3 \mapsto [0,1]$, and $\epsilon, \lambda > 0$, find a point $x \in [0,1]^3$ such that $p(g(x)) \geq p(x) - \epsilon$, or $|f(x) - x| < \epsilon$, or else two points $x, x' \in [0,1]^3$ violating the λ-Lipschitz continuity of f or of p.

THEOREM 2.2. *If A is a* PPAD-*complete problem, and B a* PLS-*complete one, then the following problem, which we call* EITHER SOLUTION$_{A,B}$, *is complete for* PPAD $\cap$ PLS: *Given an instance x of A and an instance y of B, find either a solution of x or a solution of y.*

Proof. The problem is clearly in both PLS and PPAD, because it can be reduced to both A and B. To show completeness, consider any problem C in PPAD $\cap$ PLS. Since A is PPAD-complete and C is in PPAD, there is a reduction such that, given an instance x of C, produces an instance $f(x)$ of A, such that from any solution of $f(x)$ we can recover a solution of C. Similarly, there is a reduction g from C to B. Therefore, going from x to $(f(x), g(x))$ is a reduction from C to EITHER SOLUTION$_{A,B}$.

COROLLARY 2.1. EITHER FIXPOINT *is complete for* PPAD $\cap$ PLS.

Proof. It follows from Theorem 2.2, given Theorem 2.1 and the fact that BROUWER FIXPOINT is PPAD-complete.

2.1 The Class CLS. We introduce our new class, a subset of PPAD ∩ PLS, by way of the problem, CONTINUOUS LOCALOPT, which is a special case of REAL LOCALOPT where both functions are continuous: Given two arithmetic circuits computing functions $f : [0,1]^3 \to [0,1]^3$ and $p : [0,1]^3 \to [0,1]$, and constants $\epsilon, \lambda > 0$, find a point x such that $p(f(x)) \geq p(x) - \epsilon$, or two points that violate the λ-continuity of *either f or p*. We define CLS, for *Continuous Local Search*, as the class of all problems that can be reduced to CONTINUOUS LOCALOPT. In other words, CLS contains all problems seeking an approximate local optimum (more generally, an approximate stationary or KKT point) of a *continuous* function p, helped by a *continous* oracle—f—that takes you from any non-stationary point to a better one; recall that gradient descent and Newton iteration are standard such oracles when p is smooth.

THEOREM 2.3. CLS $\subseteq$ PPAD $\cap$ PLS.

Proof. We show that CONTINUOUS LOCALOPT lies in PPAD ∩ PLS. That CONTINUOUS LOCALOPT is in PLS follows from the fact that it is a special case of REAL LOCALOPT. To show that it is also in PPAD, we provide a reduction to BROUWER FIXPOINT. We reduce an instance $(f, g, \lambda, \epsilon)$ of CONTINUOUS LOCALOPT to the instance $(f, \lambda, \frac{\epsilon}{\lambda})$ of BROUWER FIXPOINT. If on this instance BROUWER FIXPOINT returns a pair of points violating f's Lipschitz continuity, we return this pair of points as a witness of this violation. Otherwise, BROUWER FIXPOINT returns a point x such that $|f(x) - x| < \frac{\epsilon}{\lambda}$. In this case, we check whether $|p(f(x)) - p(x)| \leq \lambda |f(x) - x|$. If this is not the case, the pair of points $f(x)$ and x witness the non-Lipschitz continuity of p. Otherwise, we have that $p(f(x)) \geq p(x) - \epsilon$, which is a solution to CONTINUOUS LOCALOPT.

Figure 2 shows the structure of TFNP with CLS included..

3 The Problems

Approximate Fixpoint of a Contraction Map (CONTRACTIONMAP). We are given a function $f : [0,1]^n \mapsto [0,1]^n$ and some $c = c(n) < 1$, with the associated claim that f is c-contracting (i.e. that, for all x, y, $|f(x) - f(y)| \leq c \cdot |x - y|$), and we seek an approximate fixpoint of this function, or a violation of contraction. (The choice of norm $|\cdot|$ is not important.) This problem seems easier than Brouwer's (if for no other reason, since the existence proof is
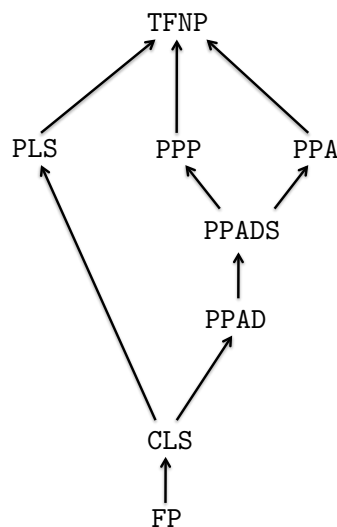

Figure 2: The location of CLS inside TFNP.

much simpler [1]), but there is no known polynomial-time algorithm (but there is a pseudo-PTAS based on a recent result in Analysis, see [2]).

Linear Complementarity Problem for P-matrices (P-LCP). In the linear complementarity problem (LCP) we are given an $n \times n$ matrix M and an n-vector q, and we seek two vectors x, y such that

$$y = Mx + q, \ (x, y) \geq 0 \text{ and } x^T y = 0.$$

LCP generalizes linear programming, and is an active research area; it is in general NP-complete, but enough special cases are in P and in PPAD. P-LCP is the spacial case when M is a P-matrix, i.e. if all its principal minors are positive; it is known to be in PPAD [19].

Finding a stationary point of a polynomial (KKT). We are given a polynomial p on $[0,1]^n$ and constants $\epsilon, \kappa > 0$. We are looking for a point $x \in [0,1]^n$ such that

$$(3.1) \quad p(y) - p(x) \leq \frac{c}{2} \cdot \epsilon^2 + \kappa, \forall y \in B_2(x, \epsilon) \cap [0,1]^n,$$

where $B_2(x, \epsilon)$ represents the ℓ_2 ball of radius ϵ around x and c is the Lipschitz constant of the gradient of p with respect to the ℓ_2 norm (since p is a polynomial, the coordinates of its gradient are also polynomials and thus Lipschitz continuous on $[0,1]^n$). Notice that all true stationary points of p satisfy (3.1), by Taylor's Theorem. Moreover, requiring (3.1), instead that the gradient of p be zero, makes the problem total. Indeed, a polynomial may not have a stationary point in $[0,1]^n$ (e.g., consider $p(x) = x$), but even when it does not, there is still a point at the boundary with small margin

of increase within the hypercube as required by (3.1) (and another one with small margin of decrease). A proof of totality is given within the proof of Theorem 3.1 below.

Simple Stochastic Games (SSGs**).** Few problems in the fringes of P have attracted as much interest as SSGs, proposed by Shapley [23] and studied in earnest from the computational point of view starting with Anne Condon [5]. One of the many ways of restating the problem is this: Given a directed graph whose nodes have outdegree two and labels "min," "max," and "average," plus two sinks labeled 0 and 1, find rational values for the non-sink nodes such that the value of every min mode is the minimum of the values of its two successors, and similarly for max and average. By monotonicity, such values always exist.

Nash equilibrium in network coordination games (NetCoordination**).** We are given an undirected graph, in which each node is a player and each edge is a 2-player coordination game between its endpoints (a game in which the two players get identical payoffs for any combination of strategy choices). Each node has the same strategy set at each of the games it participates. Each player chooses a strategy, and gets the sum of the payoffs from the games on its adjacent edges. We seek a mixed Nash equilibrium of this game. Finding a pure Nash equilibrium is PLS-complete [4].

Nash equilibrium in congestion games (Congestion**).** The strategy set of each of n players is a set of subsets of E, a set of edges (resources). Each edge has a *delay function d* mapping $\{1, 2, \ldots, n\}$ to the positive integers. Once all players select a subset each from its strategy set, say $P_1, \ldots, P_n \subseteq E$, we calculate the congestion $c(e)$ of each edge e (the number of P_j's that contain e), and the (negative) payoff of player i is $-\sum_{e \in P_i} d(c(e))$. Finding a pure Nash equilibrium in such a game is known to be PLS-complete [9].

Nash equilibrium in implicit congestion games (ImplicitCongestion**).** The special case and succinct representation of the above problem in the case in which E is the set of edges of an actual network, every player is associated with two nodes in the network, and the set of strategies of player i is the set of all paths between the two points corresponding to i. It is known to be PLS-complete to find a pure Nash equilibrium [9].

THEOREM 3.1. *The following problems are in the class* CLS*:*

1. ContractionMap.

2. P-LCP.

3. KKT

4. SSG

5. NetCoordinatination

6. Congestion

7. ImplicitCongestion

Proof. 1. It is not hard to check that Contraction-Map reduces to Continuous Localopt with potential $p(x) = |f(x) - x|$, function f, Lipschitz constant $\lambda = c + 1$, and accuracy $\epsilon = (1 - c) \cdot \delta$, where δ is the desired approximation in Contraction-Map.

2. That P-LCP is in CLS follows from a powerful algorithmic technique developed [13] for P-LCP. A Newton-like iteration $f(x)$ within the region $\{x \geq 0 : y = Mx + q \geq 0, x^T \cdot y \geq \epsilon\}$ decreases the potential function $p(x) = 2n \log x^T \cdot y - \sum_{i=1}^{n} \log x_i y_i$ until $x^T \cdot y < \epsilon$; the logarithms can be approximated by their expansion. If at some point the iteration cannot be carried out because of singularity, M is not a P matrix. By taking ϵ small enough, the precise rational solution x can be deduced from the algorithm's result.

3. To show that KKT is in CLS we need to exhibit a potential p and a potential-improving map f. The potential is the given polynomial p, whereas f is a carefully constructed version of the gradient descent method. However, the details are a bit involved:

First, we can reduce our instance of KKT to an instance where the range of the input polynomial is $[0, 1]$. Indeed, for a given instance (p, ϵ, κ) of KKT, we can compute an upper bound U on the magnitude of p, e.g., by summing the absolute values of the coefficients of all the monomials, and consider the alternative polynomial $p'(x) := \frac{1}{2U} \cdot p(x) + \frac{1}{2}$. Moreover, we tweak ϵ and κ to $\epsilon' = \epsilon$ and $\kappa' = \frac{\kappa}{2U}$, so that a solution of (p', ϵ', κ') is a solution of (p, ϵ, κ).

Given this, in the remaining discussion we assume that p ranges in $[0, 1]$. Since p is a polynomial, its partial derivatives are also polynomials. So, it is easy to compute an upper bound M on the magnitude of the derivatives inside $[0, 1]^n$. Moreover, we can easily compute upper bounds λ_1 and λ_2 on the ℓ_2-ℓ_2 Lipschitz constants of p and its gradient respectively.

Next, for some η to be decided later, we define a continuous function $f : [0, 1]^n \longrightarrow [0, 1]^n$, such that

for all $x \in [0,1]^n$, $x \mapsto^f y$, where y is as follows:

$$y_i = \min\left\{1, \max\left\{0, x_i + \frac{\partial p}{\partial x_i} \cdot \eta\right\}\right\}.$$

Observe that the function f is $\sqrt{n}(1 + \eta\lambda_2)$-Lipschitz in the ℓ_2-ℓ_2 sense. For some δ to be decided later, we invoke CONTINUOUS LOCALOPT on input: function f, potential $1 - p$, approximation δ, and Lipschitz constant $\lambda = \max\{\lambda_1, \sqrt{n}(1 + \eta\lambda_2)\}$ to find a point x, such that

$$(3.2) \qquad p(f(x)) \leq p(x) + \delta.$$

(Notice that no violation of Lipschitzness will be detected by CONTINUOUS LOCALOPT.) We want to show that, when (3.2) is satisfied, x satisfies (3.1). The following useful lemma is easy to obtain using basic calculus.

LEMMA 3.1. (TAYLOR'S THEOREM) *Let* $g : \mathbb{R}^n \to \mathbb{R}$ *be a continuously differentiable and suppose that its gradient* ∇g *is c-Lipschitz in the ℓ_2-ℓ_2 sense. Then, for all $x, x_0 \in \mathbb{R}^n$:*

$$|g(x) - g(x_0) - \nabla g(x_0) \cdot (x - x_0)| \leq \frac{c}{2}|x - x_0|_2^2.$$

Using Lemma 3.1 and (3.2), we obtain (let us use the upper bound of λ_2 on c):

$$(3.3) \quad \nabla p(x) \cdot (f(x) - x) \leq \delta + \frac{\lambda_2}{2}|f(x) - x|_2^2$$
$$\leq \delta + \frac{\lambda_2}{2}n \cdot \eta^2 M^2.$$

From the definition of f it follows that:

(a) if $f(x)_i \neq 0, 1$, then $f(x)_i - x_i = \frac{\partial p}{\partial x_i} \cdot \eta$;

(b) if $f(x)_i = 0$, then $\frac{\partial p}{\partial x_i} \leq \frac{-x_i}{\eta}$;

(c) if $f(x)_i = 1$, then $\frac{\partial p}{\partial x_i} \geq \frac{1-x_i}{\eta}$.

Hence, $f(x)_i - x_i$ and $\frac{\partial p}{\partial x_i}$ always have the same sign. Using this and (3.3) we deduce the following:

(a) if $f(x)_i \neq 0, 1$, then

$$\left|\frac{\partial p}{\partial x_i}\right| \leq \sqrt{\frac{\delta + \frac{\lambda_2}{2} n \cdot \eta^2 M^2}{\eta}};$$

(b) if $f(x)_i = 0$, $(-x_i)\frac{\partial p}{\partial x_i} \leq \delta + \frac{\lambda_2}{2}n \cdot \eta^2 M^2$;

(c) if $f(x)_i = 1$, $(1 - x_i)\frac{\partial p}{\partial x_i} \leq \delta + \frac{\lambda_2}{2}n \cdot \eta^2 M^2$.

Now, for any $y \in B_2(x, \epsilon) \cap [0,1]^n$, we have from Lemma 3.1 the following:

$$p(y) \leq p(x) + \nabla p(x) \cdot (y - x) + \frac{c}{2}|y - x|_2^2.$$

We proceed to bound the term $\nabla p(x) \cdot (y - x) = \sum_{i=1}^n \frac{\partial p}{\partial x_i}(y_i - x_i)$ term-by-term.

- if $f(x)_i \neq 0, 1$, then we have from the above that

$$\frac{\partial p}{\partial x_i}(y_i - x_i)$$
$$\leq \sqrt{\frac{\delta + \frac{\lambda_2}{2} n \cdot \eta^2 M^2}{\eta}}|y_i - x_i|$$
$$\leq \sqrt{\frac{\delta + \frac{\lambda_2}{2} n \cdot \eta^2 M^2}{\eta}}\epsilon.$$

- if $f(x)_i = 0$, then $\frac{\partial p}{\partial x_i} \leq 0$, hence $\frac{\partial p}{\partial x_i}(y_i - x_i)$ will be negative unless $y_i \leq x_i$ and the quantity is maximized for $y_i = 0$. In this case:

$$\frac{\partial p}{\partial x_i}(y_i - x_i) \leq \frac{\partial p}{\partial x_i}(-x_i) \leq \delta + \frac{\lambda_2}{2}n \cdot \eta^2 M^2.$$

- if $f(x)_i = 1$, we obtain similarly that $\frac{\partial p}{\partial x_i}(y_i - x_i) \leq \delta + \frac{\lambda_2}{2}n \cdot \eta^2 M^2$.

Putting everything together, we obtain:

$$(3.4) \qquad p(y) \leq p(x) + \frac{c}{2}\epsilon^2 +$$
$$+ n \cdot \max\left\{\delta + \frac{\lambda_2 \eta^2 M^2 n}{2}, \sqrt{\frac{\delta + \frac{\lambda_2 \eta^2 M^2 n}{2}}{\eta}}\epsilon\right\}$$

Choosing $\delta = \eta^2$, and $\eta = \eta(\epsilon, n, M, \lambda_2, \kappa) > 0$ small enough we can make sure that (3.4) gives

$$p(y) \leq p(x) + \frac{c}{2} \cdot \epsilon^2 + \kappa.$$

This completes the proof of 3.

4. It is known that solving a SSG can be reduced to computing an approximate fixpoint of a contraction map (see, e.g., [8]).

5. We show that NETCOORDINATION is polynomial-time reducible to KKT. For notational simplicity, we give below the proof for two strategies per node. Suppose that the game has n players, $1, \ldots, n$, and suppose that for every pair $i < j \in [n]$, we are given a 2×2 payoff table $A^{(i,j)}$ specifying the common payoff that players receive from their joint interaction for different selections

of strategies. Given this representation, if the players play strategy 2 with respective probabilities $x_1, x_2, \ldots, x_n \in [0,1]$, player i receives payoff:

$$\mathcal{U}_i(x) = \sum_{j<i} \hat{x}_j^{\mathrm{T}} A^{(j,i)} \hat{x}_i + \sum_{i<j} \hat{x}_i^{\mathrm{T}} A^{(i,j)} \hat{x}_j,$$

where, for all j, $\hat{x}_j = (1 - x_j, x_j)^{\mathrm{T}}$. Let us then define the function $\Phi(x) = \frac{1}{2} \sum_i \mathcal{U}_i(x)$. We observe the following.

LEMMA 3.2. (POTENTIAL FUNCTION) *For any collection of mixed strategies x_{-i} for all players of the game except player i, and for any pair of mixed strategies x_i and x_i' for player i we have:* $\mathcal{U}_i(x_i'; x_{-i}) - \mathcal{U}_i(x_i; x_{-i}) = \Phi_i(x_i'; x_{-i}) - \Phi_i(x_i; x_{-i})$.

Clearly, $\Phi(x)$ is a multilinear polynomial on $x_1, \ldots, x_n$ of degree 2. We can easily compute an upper bound U on the Lipschitz constant of $\Phi(\cdot)$ with respect to the ℓ_2 norm over $[0,1]^n$. Then let us feed Φ into the problem KKT together with ϵ and κ satisfying the condition $\frac{U}{2}\epsilon^3 + \kappa \cdot \epsilon \leq \delta$, for some δ to be chosen later. After solving KKT we obtain a point x such that

$$(3.5) \quad \Phi(y) - \Phi(x) \leq \frac{U}{2} \cdot \epsilon^2 + \kappa$$
$$\leq \frac{\delta}{\epsilon}, \forall y \in B_2(x, \epsilon) \cap [0,1]^n.$$

Using the linearity of Φ with respect to each player's mixed strategy, we can show that the point x is a δ-approximate Nash equilibrium of the coordination game (i.e. a collection of mixed strategies such that no player can improve her payoff by unilaterally changing her mixed strategy). If we choose δ sufficiently small, the approximate equilibrium can be rounded to an exact equilibrium via Linear Programming [7].

6. CONGESTION can be reduced to KKT, by adapting the previous proof. Now player i's mixed strategy $\mathcal{X}_i$ is a distribution over subsets of E. For a collection of mixed strategies $\mathcal{X}_1, \ldots, \mathcal{X}_n$, we can define the function $\Phi(\mathcal{X}) := \mathbb{E}_{S \sim \mathcal{X}} \phi_R(S)$, where $\phi_R(\cdot)$ is Rosenthal's potential function, showing that congestion games always have pure Nash equilibria [21]. It can be established that $\Phi(\cdot)$ is an exact potential function of the congestion game, extending Rosenthal's potential function from pure to mixed strategies strategies. Moreover, Φ is a multilinear function on the players' mixed strategies. Since Φ is an exact potential function and multilinear, finding an approximate mixed Nash equilib-

rium can be reduced to KKT,[2] for the same reason that NETCOORDINATION is reducible to KKT.

7. That IMPLICITCONGESTION is in CLS can be shown in a similar way, except that now a variant of KKT is required, where the polynomial is defined over $\times_{i=1}^n \mathcal{F}_i$, where $\mathcal{F}_i$ is the flow polytope from which player i choses a mixed strategy. Again the polynomial used is the expectation of Rosenthal's potential function over the paths sampled from the players' mixed strategies. The details are omitted.

4 The Class CCLS

The four last problems in the list above (NETCOORDINATION, CONGESTION, IMPLICITCONGESTION, and SSG), lie in another interesting subclass of PPAD $\cap$ PLS, which we call CCLS for convex CLS. Let p be a continuous function from $\prod_i D_i \to \Re_+$, where for $i = 1, \ldots, n$, D_i is a convex domain such as $[0,1]^k$ or the k-simplex $\Delta^k \subset \mathbb{R}^{k+1}$. To simplify our discussion, let us assume that all the D_i's are k-simplices. and we write the function as $p(x_1, \ldots, x_n)$, where each $x_i \in \Delta^k$. We call p *componentwise convex* if for every $i \leq n$ and every fixed set of values $v_j \in D_j$ for $j \neq i$, the function $p(v_1, \ldots, v_{i-1}, x_i, v_{i+1}, \ldots, v_n)$ of x_i is convex. Notice that a componentwise convex function may not be convex — in particular, it can have many local minima. We say that a point $x_1, \ldots, x_n$ is a (δ, ϵ)-*componentwise local-minimum* iff for all $i \leq n$, $j \leq k+1$:

$$(4.6) \quad p((1-\delta)x_i + \delta e_j \; ; \; x_{-i}) \geq p(x_i \; ; \; x_{-i}) - \epsilon.$$

In other words, mixing any single component of x with a unit vector along some dimension j with mixing weight δ, decreases the value of f by at most ϵ. Given this definition, we define COMPONENTWISECONVEX LOCALMIN as the following problem: Given a (purported) c-Lipschitz continuous componentwise convex function p and $\epsilon, \delta > 0$, find a (δ, ϵ)-approximate local minimum of p, or three points that violate componentwise convexity, or two points violating the c-Lipschitzness. We define CCLS as all problems reducible to COMPONENTWISECONVEX LOCALMIN. We show

THEOREM 4.1. *1.* CCLS $\subseteq$ PPAD $\cap$ PLS.

2. NETCOORDINATION, CONGESTION, IMPLICITCONGESTION, $SSG \in$ CCLS.

Proof. **(Sketch.)** The proof that CCLS $\subseteq$ PLS is not hard as the problem at hand is a local search problem (see Appendix A.3). To show that CCLS $\subseteq$ PPAD

---

[2]in particular, its generalization where the function is defined over the product of larger simplices

we reduce the local search to a fixpoint computation problem. Intuitively, the d components of the domain of p can be thought of as players of an exact potential game (with potential p) seeking their own utility, while interacting with others in complex ways. And the goal is to find a Nash equilibrium of the game. Inspired by this we define a smoothened version of the function that Nash used in his paper [17] and reduce the problem of finding a Nash equilibrium of the game to BROUWER. The proof that this construction works is quite intricate and postponed to Appendix A.3.

For the second part of the theorem, the first three problems are in CCLS by virtue of the potential functions used to prove that they lie in CLS. In particular, these potential functions are componentwise linear, and their componentwise local maxima correspond to Nash equilibria. For SSG, we utilize a proof due to Condon, whereby SSG can be formulated as the local minimum of a quadratic function in a convex polytope. Even though the quadratic function is not componentwise convex, it can be modified to incorporate a penalty for leaving the polytope, and the resulting function is indeed componentwise convex.

5 On Exact Fixpoints, and Variants of our Classes

As Etessami and Yannakakis pointed out, interesting complexity phenomena start happening when one seeks exact, not approximate, Brouwer fixpoints [8]. Since in this paper we are also interested in fixpoint-like computations, it would be interesting to ponder the complexity of the exact versions of the problems CONTINUOUS LOCALOPT and COMPONENTWISECONVEX LOCALMIN. We make the following remark in this connection:

REMARK 5.1. *One interesting fact about FIXP, the exact version of* PPAD, *is that the* SUMOFSQUAREROOTS *problem (given integers $a_1, \ldots, a_n, b$, is $\sum_{i=1}^{n} \sqrt{a_i} > b$?), not known to lie in* NP, *can be reduced to it. It turns out that it can also be reduced to the exact version of* CCLS! *(The same is true of* CLS.*) Consider the polynomial:*

$$p(x, t) = \frac{t+1}{2} \cdot \left(\frac{3x_1}{2} - \frac{x_1^3}{2a_1} + \ldots + \frac{3x_n}{2} - \frac{x_n^3}{2a_n} - b \right)$$

where $t \in [0, 1]$ and $x_i \in [0, a_i]$, for all i. It is not hard to check that $p(x, t)$ is component-wise concave. Moreover, it has a unique exact component-wise local maximum in which $x_i = \sqrt{a_i}$, for all i, and, if $\sum_i \sqrt{a_i} > b$, then $t = 1$, while, if $\sum_i \sqrt{a_i} < b$, $t = 0$. Given that we can decide in polynomial time whether $\sum_i \sqrt{a_i} = b$, it follows from the above discussion that deciding whether $t = 1$ at the exact component-wise local maximum of p is SUMOFSQUAREROOTS-*hard.*

Moreover, recall that PLS is equivalent to REAL LO-CAOPT, which defined in terms of a continuous potential function p and a non-necessarily continuous neighborhood function f. In REAL LOCALOPT we seek an approximate fixed point of f with respect to p. Below we discuss the complexity of variants of REAL LOCALOPT, when we either seek exact solutions, or we drop the continuity assumption on the potential.

REMARK 5.2. *Given two functions $f : [0, 1]^3 \to [0, 1]^3$ and $p : [0, 1]^3 \to [0, 1]$, where p is continuous but f is not necessarily continuous, there always exists a point x such that $p(f(x)) \geq p(x)$. Indeed, since p is continuous on a compact (i.e. closed and bounded) subset of the Euclidean space it achieves its minimum at a point $x^* \in [0, 1]^3$. Regardless then of the value of $f(x^*)$ it must be that $p(f(x^*)) \geq p(x^*)$. At the same time, it follows from our discussion in Remark 5.1 that the exact version of* REAL LOCALOPT *is also* SUMOFSQUAREROOTS-*hard (and therefore, as far as we know, may well be outside of* NP*).*

REMARK 5.3. *Given two functions f, $p : [0, 1] \to [0, 1]$ that are not continuous, there may not be a point x such that $p(f(x)) \geq p(x)$. To see this, consider these functions: $(f, p : [0, 1] \mapsto [0, 1]$ with $f(x) = p(x) = \frac{x}{2}$ if $x = 2^{-i}, i = 0, 1, 2, \ldots$ and $f(x) = p(x) = 1$ otherwise. Note however that these functions cannot be computed by an arithmetic circuit (a loop is needed); it is an interesting question whether an example of this sort exists that is so computable.*

6 Conclusions and Open Problems

Two decades ago, the definition of PPAD was an invitation to study fixpoint problems from the point of view of Complexity; its true dividend came fifteen years later, with the proof that BROUWER and NASH are the same problem. Here we have pointed out that there are rich Complexity considerations in a realm with many important and fascinating problems, that seems to lie intriguingly close to P.

There is a plethora of open problems left by the ideas introduced in this paper, of which the most obvious:

- It would be very interesting to identify natural CLS-complete problems. CONTRACTIONMAP, P-LCP and KKT are prime suspects.

- Is CCLS ⊆ CLS? We conjecture that this is the case.

- Separate by oracles PPAD ∩ PLS from CLS, and CLS from CCLS (one or both ways).

- Are there PTAS's for problems in CLS beyond CONTRACTIONMAP? It would be fascinating if

there is a general-purpose PTAS. Indeed, there is a PTAS for certain special cases of KKT [26].

Acknowledgment. We are grateful to Nimrod Megiddo, Rahul Savani, Steve Vavasis, and Yinyu Ye for several illuminating discussions on these problems.

References

[1] S. Banach. Sur Les Opérations Dans Les Ensembles Abstraits et Leur Application aux Équations Intégrales. *Fund. Math.* 3: 133–181, 1922.

[2] J.-B. Baillon, R. E. Bruck. The Rate of Asymptotic Regularity is $O(1/\sqrt{n})$. *Theory and Applications of Nonlinear Operators of Accretive and Monotone Types,* Lecture Notes in Pure and App. Math., 178:51–81, Dekker, New York, 1996.

[3] P. Beame, S. Cook, J. Edmonds, R. Impagliazzo and T. Pitassi. The Relative Complexity of NP Search Problems. *Journal of Computer and System Sciences*, 57(1):13–19, 1998.

[4] Y. Cai and C. Daskalakis. On Minmax Theorems for Multiplayer Games. *In the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 2011.

[5] A. Condon. The Complexity of Stochastic Games. *Information and Computation,* 96(2): 203–224, 1992.

[6] X. Chen and X. Deng. Settling the Complexity of 2-Player Nash-Equilibrium. *In the 47th Annual IEEE Symposium on Foundations of Computer Science,* FOCS 2006.

[7] C. Daskalakis, P. W. Goldberg and C. H. Papadimitriou. The Complexity of Computing a Nash Equilibrium. *In the 38th ACM Symposium on Theory of Computing,* STOC 2006.

[8] K. Etessami and M. Yannakakis. On the Complexity of Nash Equilibria and Other Fixed Points. *In the 48th Annual IEEE Symposium on Foundations of Computer Science,* FOCS 2007.

[9] A. Fabrikant, C.H. Papadimitriou and K. Talwar. The Complexity of Pure Nash Equilibria. *In the 36th ACM Symposium on Theory of Computing,* STOC 2004.

[10] M. R. Garey, R. L. Graham and D. S. Johnson. Some NP-complete Geometric Problems. *In 8th ACM Symposium on Theory of Computing,* STOC 1976.

[11] D. S. Johnson, C. H. Papadimitriou and M. Yannakakis. How Easy is Local Search? *Journal of Computer and System Sciences*, 37(1): 79–100, 1988.

[12] B. Juba. On the Hardness of Simple Stochastic Games. Master's thesis, CMU, 2005.

[13] M. Kojima, N. Megiddo and Y. Ye. An Interior Point Potential Reduction Algorithm for the Linear Complementarity Problem. *Mathematical Programming*, 1992.

[14] M. Kojima, N. Megiddo, T. Noma, A. Yoshise. *A Unified Approach to Interior Point Algorithms for Linear Complementarity Problems.* Lecture Notes in Computer Science 538, Springer-Verlag, 1991.

[15] N. Megiddo. A Note on the Complexity of P-matrix LCP and Computing an Equilibrium. *Research Report RJ 6439, IBM Almaden Research Center,* 1988.

[16] K. G. Murty and S. N. Kabadi. Some NP Complete Problems in Quadratic and Nonlinear Programming. *Mathematical Programming,* 39:117–129, 1987.

[17] J. Nash. Non-cooperative games. *Annals of mathematics,* 1951.

[18] C. H. Papadimitriou. *Computational Complexity*, Addison Wesley, 1994.

[19] C. H. Papadimitriou. On the Complexity of the Parity Argument and Other Inefficient Proofs of Existence. *Journal of Computer and System Sciences*, 48(3): 498–532, 1994.

[20] C. H. Papadimitriou. The Euclidean Traveling Salesman Problem is NP-Complete. *Theor. Comp. Sci.,* 4:237–244, 1977.

[21] R. W. Rosenthal. A Class of Games Possessing Pure-Strategy Nash Equilibria. *International Journal of Game Theory,* 2:6567, 1973.

[22] A. A. Schäffer and M. Yannakakis. Simple Local Search Problems that are Hard to Solve. *SIAM J. Comput.,* 20(1): 56–87, 1991.

[23] L. S. Shapley. Stochastic Games. *In Proceedings of the National Academy of Sciences,* 39:1095–1100, 1953.

[24] D. Solow, R. Stone and C. A. Tovey. Solving LCP on P-matrices is probably not NP-hard. *Unpublished Note,* November 1987.

[25] P. Tiwari. A Problem that is Easier to Solve on the Unit-Cost Algebraic RAM. *Journal of Complexity,* 393–397, 1992.

[26] Y. Ye. On the Complexity of Approximating a KKT Point of Quadratic Programming. *Mathematical Programming,* 80:195–212 1998.

A Omitted Details from Section 4

For reasons related to our proof techniques, it is more convenient to think in terms of (componentwise) concavity and (componentwise) local maxima, instead of convexity and local minima. The corresponding optimization problem is called ComponentwiseConcave LocalMax. Clearly, ComponentwiseConvex LocalMin and ComponentwiseConcave LocalMax are equivalent, so this change is without loss of generality.

A.1 Some Notation.

DEFINITION A.1. (SIMPLEX) *For an integer $k > 0$, we define the k-simplex Δ^k as follows*

$$\Delta^k := \left\{ (x_1, \ldots, x_{k+1}) \in \mathbb{R}_+^{k+1} \;\middle|\; \sum_{i=1}^{k+1} x_i = 1 \right\}.$$

DEFINITION A.2. (INDEXING, DISTANCES) *We represent an element $x \in \prod_{i=1}^n S_i$, where $S_i \subseteq \mathbb{R}^{n_i}$, as*

$x = (x_1, x_2, \ldots, x_n)$ where, for all $i \in [n]$, $x_i \in S_i$. We call x_i the i-the component of x. The value of the j-th coordinate of x_i is denoted by $x_i(j)$. Moreover, for all $i \in [n]$, we denote by x_{-i} the vector comprising of all but the i-th component of x, and we also use the convenient notation $(x_i \ ; \ x_{-i}) \equiv x$. Finally, for $x, y \in \prod_{i=1}^n S_i$, we define

$$|x - y|_\infty = \max_{i \in [n]} \{|x_i - y_i|_\infty\}.$$

DEFINITION A.3. (COMPONENT-WISE LOCAL MAXIMUM) *Given a function* $f : \prod_{i \in [n]} S_i \longrightarrow \mathbb{R}$, *where* $S_i := \Delta^{n_i - 1}$, *a point* x *is called a* component-wise (δ, ϵ)-local maximum *of* f *iff, for all* $i \in [n]$, $j \in [n_i]$:

$$(1.7) \qquad f((1-\delta)x_i + \delta e_j \ ; \ x_{-i}) - f(x_i \ ; \ x_{-i}) \leq \epsilon.$$

In other words, a point x *is a component-wise approximate local maximum of* f, *if mixing any single component of* x *with a unit vector using mixing weight* δ, *increases the value of* f *by at most* ϵ.

DEFINITION A.4. (COMPONENT-WISE CONCAVITY) *A function* $f : \prod_{i \in [n]} S_i \longrightarrow \mathbb{R}$, *where* S_i *is a convex set, is called* component-wise concave *iff for all* $i \in [n]$, $x_{-i} \in \prod_{j \neq i} S_j$, $x_i, x_i' \in S_i$ *and* $\alpha \in [0, 1]$:

$$f(\alpha x_i + (1-\alpha)x_i' \ ; \ x_{-i}) \geq$$
$$\alpha f(x_i \ ; \ x_{-i}) + (1-\alpha)f(x_i' \ ; \ x_{-i}).$$

A.2   Smoothened Nash Function. As a basic tool to reduce local optimization to fixed point computation, we employ the following construction inspired by Nash's theorem [17].

DEFINITION A.5. (SMOOTHENED NASH FUNCTION) *Let* $f : S \to \mathbb{R}$, *where* $S = \prod_{i=1}^n S_i$, $S_i = \Delta^{n_i - 1}$. *For* $\delta > 0$, *we define the function* $g_\delta : S \to S$ *based on* f *as follows: For* $x \in S$, *the function* g_δ *maps* $x \mapsto^{g_\delta} y$, *where for all* $i \in [n]$ *and* $j \in [n_i]$:

$$y_i(j) = \frac{x_i(j) + \max\{0, I_{ij}^\delta\}}{1 + \sum_k \max\{0, I_{ik}^\delta\}}.$$

where I_{ij}^δ *represents the improvement on the value of* $f(x)$ *by mixing* x_i *with* e_j *while keeping the other components of* x *fixed, i.e.,*

$$I_{ij}^\delta := f((1-\delta)x_i + \delta e_j \ ; \ x_{-i}) - f(x_i \ ; \ x_{-i}).$$

We show that the smoothened Nash function corresponding to some function f inherits Lipschitzness properties from f. The proof of the following is omitted.

LEMMA A.1. *Suppose that* $f : S \to \mathbb{R}$, *where* $S = \prod_{i=1}^n S_i$, $S_i = \Delta^{n_i - 1}$, *is* c-Lipschitz with respect to the

ℓ_∞ *norm on* S, *i.e., for all* $x, y \in S$, $|f(x) - f(y)| \leq c \cdot |x - y|_\infty$. *Then, for all* $\delta > 0$, *there exists* $c' = c'(n, \max\{n_i\}, c, \delta) > 0$, *such that the smoothened Nash function corresponding to* f *(see Defintion A.5) is* c'-Lipschitz, *i.e., for all* $x, y \in S$:

$$|g_\delta(x) - g_\delta(y)|_\infty \leq c' \cdot |x - y|_\infty.$$

Moreover, the description complexity of c' *is polynomial in the description complexity of* n, $\max\{n_i\}$, c, *and* δ.

A.3   Proof of CCLS $\subseteq$ PPAD $\cap$ PLS. The central element of our proof is the following result, establishing that the fixed points of the smoothened Nash function corresponding to a componentwise concave function, are componentwise local maxima.

THEOREM A.1. *Let* $f : S \to [0, f_{\max}]$, *where* $S = \prod_{i=1}^n S_i$ *and* $S_i \equiv \Delta^{n_i - 1}$, *for all* i, *be continuous and component-wise concave. For all* $\delta \in (0, 1)$, $\epsilon > 0$, *there exists some* $\zeta = \zeta(\epsilon, f_{\max}, \max\{n_i\}) > 0$ *such that if* $|g_\delta(x) - x|_\infty \leq \zeta$, *where* g_δ *is the smoothened Nash function corresponding to* f *(see Definition A.5), then* x *is a component-wise* (δ, ϵ)-local maximum of f. *Moreover, the description complexity of* ζ *is polynomial in the description complexity of* ϵ, $f_{\max}$ *and* $\max\{n_i\}$.

Proof of Theorem A.1: Suppose that x satisfies $|g_\delta(x) - x|_\infty \leq \zeta$ for some $\zeta \in (0, 1)$. Let us fix some $i \in [n]$ and assume, without loss of generality, that

$$(1.8)$$
$$f_{i1}^\delta(x) \geq \ldots \geq f_{ik}^\delta(x) > f(x) \geq f_{ik+1}^\delta(x) \geq \ldots \geq f_{in_i}^\delta(x),$$

where $f_{ij}^\delta(x) := f((1-\delta)x_i + \delta e_j \ ; \ x_{-i})$. For all $j \in [n_i]$, observe that $|g_\delta(x)_i - x_i|_\infty \leq \zeta$ implies

$$x_i(j) \sum_{k \in [n_i]} \max\{0, I_{ik}^\delta\}$$

$$\leq \max\{0, I_{ij}^\delta\} + \zeta \left(1 + \sum_{k \in [n_i]} \max\{0, I_{ik}^\delta\}\right).$$

Setting $\zeta'' := \zeta(1 + n_i f_{\max})$, the above inequality implies

$$(1.9) \qquad x_i(j) \sum_{k \in [n_i]} \max\{0, I_{ik}^\delta\} \leq \max\{0, I_{ij}^\delta\} + \zeta''.$$

Let us then define $t := x_i(k+1) + x_i(k+2) + \ldots + x_i(n_i)$ and distinguish the following cases

1. If $t \geq \frac{\sqrt{\zeta''}}{f_{\max}}$, then summing Equation (1.9) over $j = k+1, \ldots, n_i$ implies

$$t \sum_{k \in [n_i]} \max\{0, I_{ik}^\delta\} \leq (n_i - k)\zeta'',$$

which gives

(1.10)
$$\max_k\{\max\{0, I_{ik}^\delta\}\} \leq \sum_{k \in [n_i]} \max\{0, I_{ik}^\delta\} \leq n_i\sqrt{\zeta''}f_{\max}.$$

2. If $0 < t < \frac{\sqrt{\zeta''}}{f_{\max}}$, then multiplying Equation (1.9) by $x_i(j)$ and summing over $j = 1, \ldots, n_i$ gives

(1.11)
$$\sum_{j \in [n_i]} x_i(j)^2 \sum_{k \in [n_i]} \max\{0, I_{ik}^\delta\}$$
$$\leq \sum_{j \in [n_i]} x_i(j) \max\{0, I_{ij}^\delta\} + \zeta''.$$

Now observe that for any setting of $x_i(j)$, $j \in [n_i]$, it holds that

(1.12)
$$\sum_{j \in [n_i]} x_i(j)^2 \geq \frac{1}{n_i}.$$

Moreover, observe that,

$$x_i = \sum_{j \in [n_i]} x_i(j)\left((1-\delta)x_i + \delta e_j\right).$$

Hence, by component-wise concavity it holds that

(1.13)
$$f(x) \geq \sum_{j \in [n_i]} x_i(j) \cdot f\left((1-\delta)x_i + \delta e_j;\ x_{-i}\right).$$

Equivalently:

$$\sum_{j \in [n_i]} x_i(j) \cdot f(x)$$
$$\geq \sum_{j \in [n_i]} x_i(j) \cdot f\left((1-\delta)x_i + \delta e_j;\ x_{-i}\right) \iff$$

(1.14)
$$\sum_{j > k} x_i(j) \cdot \left(f(x) - f\left((1-\delta)x_i + \delta e_j;\ x_{-i}\right)\right) \geq$$
$$\sum_{j \leq k} x_i(j) \cdot \left(f\left((1-\delta)x_i + \delta e_j;\ x_{-i}\right) - f(x)\right)$$

Using (1.8) and applying (1.14) and (1.12) to (1.11) we obtain

$$\frac{1}{n_i} \sum_{k \in [n_i]} \max\{0, I_{ik}^\delta\}$$
$$\leq \sum_{j > k} x_i(j) \cdot \left(f(x) - f\left((1-\delta)x_i + \delta e_j\right)\right) + \zeta''$$
$$\leq t \cdot f_{\max} + \zeta'',$$

which implies

(1.15)
$$\max_k\{\max\{0, I_{ik}^\delta\}\}$$
$$\leq \sum_{k \in [n_i]} \max\{0, I_{ik}^\delta\}$$
$$\leq n_i(t \cdot f_{\max} + \zeta'')$$
$$\leq n_i(\sqrt{\zeta''} + \zeta'') \leq 2n_i\sqrt{\zeta''},$$

assuming $\zeta'' < 1$.

Combining (1.10) and (1.15), we have the following uniform bound

$$\max_k\{\max\{0, I_{ik}^\delta\}\} \leq 2n_i\sqrt{\zeta''}\max\{f_{\max}, 1\}$$
$$= 2n_i\sqrt{\zeta(1 + n_i f_{\max})}\max\{f_{\max}, 1\}.$$

To conclude the proof of the theorem notice first that, if for all $i \in [n]$:

$$\max_{j \in [n_i]}\{\max\{0, I_{ij}^\delta\}\} \leq 2n_i\sqrt{\zeta(1 + n_i f_{\max})}\max\{f_{\max}, 1\},$$

then x is a component-wise

$$(\delta, 2\tilde{n}\sqrt{\zeta(1 + \tilde{n}f_{\max})}\max\{f_{\max}, 1\})\text{-local}$$

maximum of f, where $\tilde{n} = \max\{n_i\}$. To make $2\tilde{n}\sqrt{\zeta(1 + \tilde{n}f_{\max})}\max\{f_{\max}, 1\} \leq \epsilon$ is is sufficient to choose:

$$\zeta \leq \epsilon^2 \cdot \frac{1}{4\tilde{n}^2(1 + \tilde{n}f_{\max})\max\{f_{\max}, 1\}^2},$$

where $\tilde{n} = \max\{n_i\}$. $\blacksquare$

Using Theorem A.1 we can establish our result.

THEOREM A.2. COMPONENTWISECONCAVE LOCAL-MAX $\in$ PPAD.

Proof. To argue that COMPONENTWISECONCAVE LOCALMAX $\in$ PPAD, we use the smoothened Nash function g_δ corresponding to f (see Definition A.5). It follows from Lemma A.1 that, assuming that f is c-Lipschitz, we can efficiently compute some c' of description complexity polynomial in c, n, $\max\{n_i\}$, and δ such that g_δ is c'-Lipschitz. Now let us choose ζ as required by Theorem A.1. With a PPAD computation, we can find an approximate fixed point of g_δ, i.e. some x such that

(1.16)
$$|g_\delta(x) - x|_\infty < \zeta,$$

or a pair of points x, y that violate the c'-Lipschitzness of g_δ, i.e.

$$|g_\delta(x) - g_\delta(y)|_\infty > c' \cdot |x - y|_\infty.$$

In the latter case, using x, y we can efficiently compute a pair of points x' and y' violating the c-Lipschitzness of f (we omit the details). In the former case, we can check if x is a component-wise (δ, ϵ)-local maximum of f. If it is not, then some component x_i of x violates (1.7) despite the fact that x is an approximate fixed point. It follows from the proof of Theorem A.1 that for this to happen we must fall into Case 2 of the proof of Theorem A.1. And in this situation in order for (1.16) to fail to imply (1.7) it must be that the concavity condition (1.13) is violated at x by some component x_i. Hence, in this case we can provide a certificate that the concavity condition is violated by f.

THEOREM A.3. COMPONENTWISECONCAVE LOCAL-MAX $\in$ PLS.

Proof. Let us consider the discretization of Δ^{n_i-1} induced by the regular grid of size $k \in \mathbb{N}_+$ for some k to be decided later:

$$\Delta^{n_i-1}(k) = \left\{ x \in \Delta^{n_i-1} \;\middle|\; \begin{array}{l} x = \left(\frac{y_1}{k}, \frac{y_2}{k}, \ldots, \frac{y_{n_i}}{k}\right), \\ y_j \in \mathbb{N}_0 \text{ and } \sum_j y_j = k \end{array} \right\}.$$

Similarly, let us consider the discretization of $S := \prod_{i \in [n]} \Delta^{n_i-1}$ defined by

$$S(k) := \prod_{i \in [n]} \Delta^{n_i-1}(k).$$

Finally, let us impose the following neighborhood relation on the points of $S(k)$. Two points $x, x' \in S(k)$ are neighbors iff there exists $i \in [n]$ such that $x_j = x'_j$, for all $j \neq i$, and $\exists q \in [n_i]$ such that

$$x'_i = \text{Round}_i((1-\delta)x_i + \delta e_q)$$

or

$$x_i = \text{Round}_i((1-\delta)x'_i + \delta e_q),$$

where Round_i is the poly-time deterministic procedure described in Figure 3, which rounds a point in Δ^{n_i-1} to a point in $\Delta^{n_i-1}(k)$ that lies within ℓ_∞ distance at most $\frac{2}{k}$.[3]

Given $S(k)$ and its poly-time computable neighborhood relation, we can perform a PLS computation to locate a local maximum x^* of f.[4] The resulting point x^* satisfies the following: for all $i \in [n]$, $q \in [n_i]$:

$$f(x^*) \geq f(\text{Round}_i((1-\delta)x_i^* + \delta e_q \;;\; x_{-i}^*)).$$

---

[3]To see this observe the following: clearly, for all $j < n_i$, $|z_j - z'_j| < 1/k$; moreover, $0 \leq \epsilon_j < \frac{2}{k}$, for $j = 1, \ldots, n_i$, hence $|z_{n_i} - z'_{n_i}| < 2/k$.

[4]Notice that, unlike the definition of PLS provided in Section 1, where the neighborhood of every candidate solution comprised of a single neighbor, in the definition used here every candidate solution has many (although a polynomially-bounded number of) neighbors. The two definitions of PLS are equivalent.

Round_i: Given $z = (z_1, z_2, \ldots, z_{n_i}) \in \Delta^{n_i-1}$ define $z' = (z'_1, z'_2, \ldots, z'_{n_i}) \in \Delta^{n_i-1}(k)$ as follows:

  i. $\epsilon_1 := 0$;

  ii. for $j := 1$ to $n_i - 1$

    (a) If $\epsilon_j < \frac{1}{k}$: $z'_j = \lfloor kz_j \rfloor \cdot \frac{1}{k}$;

    (b) If $\epsilon_j \geq \frac{1}{k}$: $z'_j = \lceil kz_j \rceil \cdot \frac{1}{k}$;

    (c) $\epsilon_{j+1} := \epsilon_j + (z_j - z'_j)$;

  1. $z'_{n_i} = z_{n_i} + \epsilon_{n_i}$.

Figure 3: Procedure Round_i.

Hence, if f is c-Lipschitz then x^* satisfies the following: for all $i \in [n]$, $q \in [n_i]$:

$$f(x^*) \geq f((1-\delta)x_i^* + \delta e_q \;;\; x_{-i}^*) - c\frac{2}{k}.$$

So a choice of $k \geq \frac{2c}{\epsilon}$ guarantees that x^* is a component-wise (δ, ϵ)-local maximum of f. If x^* fails to be a (δ, ϵ)-local maximum of f this is due to an easily detectable violation of the c-Lipschitzness of f. Indeed, there must be some $i \in [n]$ and $q \in [n_i]$ such that

$$\left| f(\text{Round}_i((1-\delta)x_i^* + \delta e_q \;;\; x_{-i}^*) \right.$$
$$\left. - f((1-\delta)x_i^* + \delta e_q \;;\; x_{-i}^*) \right|$$
$$> c\frac{2}{k}$$
$$\geq d_\infty \Big(\text{Round}_i((1-\delta)x_i^* + \delta e_q \;;\; x_{-i}^*),$$
$$((1-\delta)x_i^* + \delta e_q \;;\; x_{-i}^*) \Big).$$

COROLLARY A.1. COMPONENTWISECONCAVE LOCAL-MAX $\in$ PPAD$\cap$PLS.

A.4 An Extension to Quasi-Concave Functions.
Let us define a weaker notion of component-wise concavity, called *component-wise quasi-concavity*.

DEFINITION A.6. (COMPONENT-WISE QUASI-CONCA-VITY) *A function* $f : \prod_{i \in [n]} S_i \longrightarrow \mathbb{R}$, *where* S_i *is a convex set, is called* component-wise quasi-concave *iff for all* $i \in [n]$, $x_{-i} \in \prod_{j \neq i} S_j$, $x_i, x'_i \in S_i$ *and* $\alpha \in [0, 1]$:

$$f(\alpha x_i + (1-\alpha)x'_i \;;\; x_{-i})$$
$$\geq \min(f(x_i \;;\; x_{-i}), f(x'_i \;;\; x_{-i})).$$

How hard is the componentwise local maximization problem for componentwise quasi-concave functions,

called ComponentwiseQuasiConcave LocalMax? We show that, as long as, each component of the function lives in fixed dimension the problem is still in PPAD∩PLS. Our basic tool to establish the theorem is the following generalization of Theorem A.1, whose proof provided here is much more intricate. Further details of the inclusion of ComponentwiseQuasiConcave LocalMax inside PPAD ∩ PLS are omitted.

Theorem A.4. *Let $f : S \to [0, f_{\max}]$, where $S = \prod_{i=1}^{n} S_i$ and $S_i \equiv \Delta^{n_i-1}$, for all i, be c-Lipschitz continuous with respect to the ℓ_∞ norm on S and component-wise quasi-convex. For all $\delta \in (0,1)$, $\epsilon > 0$, there exists some $\zeta = \zeta(\epsilon, f_{\max}, \max\{n_i\}, c, n) > 0$ such that, if $|g_\delta(x) - x|_\infty \leq \zeta$ where g_δ is the smoothened Nash function corresponding to f (see Definition A.5), then x is a component-wise (δ, ϵ)-local maximum of f. Moreover, the description complexity of ζ is polynomial in the description complexity of ϵ, $f_{\max}$, c and n (but not necessarily in the description complexity of $\max\{n_i\}$).*

Proof. Invoking Lemma A.1 let us call c' the Lipschitz constant of g_δ with respect to the ℓ_∞ norm on S. Let us then fix some $i \in [n]$ and throughout the argument that follows keep x_{-i} fixed. Using induction on the value of $|j \mid x_i(j) > 0| =: \ell(x_i)$, we will show that, for all $\mu \in (0,1)$ with $\mu \left(1 + (c'+1)\frac{\sqrt{1+n_i f_{\max}}}{f_{\max}}\right)^2 < 1$:

$$(1.17) \qquad |g_\delta(x)_i - x_i|_\infty \leq \mu \implies$$
$$\max_{j \in [n_i]} \{\max\{0, I_{ij}^\delta\}\} \leq \upsilon(\ell(x_i), \mu),$$

for the function $\upsilon(\ell, \mu)$ defined as follows, for $\ell \in \mathbb{N}$:

$$\upsilon(\ell, \mu) := (\mu\Phi)^{\frac{1}{2^{\ell-1}}} \cdot (1 + n_i f_{\max} + \ell\Delta) + \sqrt{\mu}\Delta,$$

where

$$\Phi := \left(1 + (c'+1)\frac{\sqrt{1+n_i f_{\max}}}{f_{\max}}\right)^2$$

and

$$\Delta := \sqrt{1 + n_i f_{\max}}\left(\frac{2c}{f_{\max}} + n_i f_{\max}\right).$$

Indeed, if $\ell(x_i) = 1$, say $x_i(1) = 1$, we have:

$$|g_\delta(x)_i(1) - x_i(1)| \leq \mu$$

$$\Rightarrow \sum_{k \in [n_i]} \max\{0, I_{ik}^\delta\} \leq \mu\left(1 + \sum_{k \in [n_i]} \max\{0, I_{ik}^\delta\}\right)$$
$$\leq \mu(1 + n_i f_{\max}).$$

(In the above we used that $I_{i1}^\delta = 0$.) Hence:

$$\max_{j \in [n_i]} \{\max\{0, I_{ij}^\delta\}\} \leq \mu(1 + n_i f_{\max}) \leq \upsilon(1, \mu).$$

Assuming now that the implication (1.17) holds for all values of $\ell(x_i) < \ell$ we show that (1.17) also holds for $\ell(x_i) = \ell$. To show this, assume first, without loss of generality, that

$$(1.18)$$
$$f_{i1}^\delta(x) \geq \ldots \geq f_{ik}^\delta(x) > f(x) \geq f_{ik+1}^\delta(x) \geq \ldots \geq f_{in_i}^\delta(x),$$

where $f_{ij}^\delta(x) := f((1-\delta)x_i + \delta e_j \; ; \; x_{-i})$. Observe that, for all $j \in [n_i]$, $|g_\delta(x)_i - x_i|_\infty \leq \mu$ implies

$$x_i(j) \sum_{k \in [n_i]} \max\{0, I_{ik}^\delta\}$$

$$\leq \max\{0, I_{ij}^\delta\} + \mu\left(1 + \sum_{k \in [n_i]} \max\{0, I_{ik}^\delta\}\right).$$

Setting $\mu'' := \mu(1 + n_i f_{\max})$, the above inequality implies

$$(1.19) \quad x_i(j) \sum_{k \in [n_i]} \max\{0, I_{ik}^\delta\} \leq \max\{0, I_{ij}^\delta\} + \mu''.$$

Let us then define $t := x_i(k+1) + x_i(k+2) + \ldots + x_i(n_i)$ and distinguish the following cases:

- If $t \geq \frac{\sqrt{\mu''}}{f_{\max}}$, then summing Equation (1.19) over $j = k+1, \ldots, n_i$ implies

$$t \sum_{k \in [n_i]} \max\{0, I_{ik}^\delta\} \leq (n_i - k)\mu'',$$

which gives

$$\max_k \{\max\{0, I_{ik}^\delta\}\} \leq \sum_{k \in [n_i]} \max\{0, I_{ik}^\delta\}$$
$$(1.20) \qquad\qquad \leq n_i \sqrt{\mu''} f_{\max} \leq \upsilon(\ell, \mu).$$

- If $0 < t < \frac{\sqrt{\mu''}}{f_{\max}}$, we use the following procedure to modify x_i. We set for all $j \in [n_i]$:

$$\hat{x}_i(j) = \begin{cases} 0, & \text{if } j \geq k+1 \\ x_i(j)/(1-t), & \text{otherwise} \end{cases}$$

The following is immediate:

Claim A.1. *The $\hat{x}_i$ resulting from the above procedure satisfies:*

$$|x_i - \hat{x}_i|_\infty \leq t.$$

Now let us define x' as follows: $x'_j = x_j$, for all $j \neq i$, and $x'_i = \hat{x}_i$. Using Claim A.1 and that g_δ is c'-Lipschitz it follows that

$$|g_\delta(x')_i - x'_i|_\infty \leq c't + \mu + t.$$

Notice that $\ell(x_i') < \ell$. Hence the induction hypothesis implies that

$$\max_{j \in [n_i]} \{\max\{0, f((1-\delta)x_i' + \delta e_j; x_{-i}) - f(x_i'; x_{-i})\}\}$$
$$\leq \upsilon(\ell - 1, \mu + (c' + 1)t).$$

Using then that f is c-Lipschitz it follows that

$$\max_{j \in [n_i]} \{\max\{0, f((1-\delta)x_i + \delta e_j; x_{-i}) - f(x_i; x_{-i})\}\}$$
$$\leq \upsilon(\ell - 1, \mu + (c' + 1)t) + 2ct$$
$$\leq \upsilon(\ell, \mu).$$

- Finally, we argue that the case $t = 0$ cannot happen. Indeed, observe that

$$x_i = \sum_{j \in [n_i]} x_i(j) \left((1-\delta)x_i + \delta e_j\right).$$

Since $t = 0$ we have that:

$$x_i = \sum_{j=1}^{k} x_i(j) \left((1-\delta)x_i + \delta e_j\right).$$

But f is component-wise quasi-concave, so that

$$f(x_i \; ; \; x_{-i}) \geq \min_{1 \leq j \leq k} \{f((1-\delta)x_i + \delta e_j; \; x_{-i})\}.$$

This contradicts (1.18), showing that the case $t = 0$ is impossible.

This completes the proof of the induction step, establishing (1.17).

To conclude the proof of the theorem notice first that, if for all $i \in [n]$:

$$\max_{j \in [n_i]} \{\max\{0, I_{ij}^{\delta}\}\} \leq \upsilon(n_i, \mu),$$

then x is a component-wise $(\delta, \max_i\{\upsilon(n_i, \mu)\})$-local maximum of f. Moreover, notice that for all ℓ, $\lim_{\mu \to 0} \upsilon(\ell, \mu) = 0$. In particular, to make $\upsilon(\ell, \mu) \leq \epsilon$ it is sufficient to choose:

$$\mu \leq \epsilon^{2^{\ell-1}} \cdot \frac{1}{\left(\Phi^{1/2^{\ell-1}}(1 + n_i f_{\max} + \ell\Delta) + \Delta\right)^{2^{\ell-1}}}.$$