On Combinatorial Structure and Algorithms for Parity Games

Henrik Björklund, Sven Sandberg & Sergei Vorobyov
Information Technology Department, Uppsala University, Sweden

Department of Information Technology
Uppsala University
Box 337, SE-751 05 Uppsala, Sweden

Technical report 2003-002
January 2003
ISSN 1404-3203

Abstract

In this paper we identify and systematically explore the combinatorial structure underlying parity and simple stochastic games. We introduce the class of Completely Local-Global (CLG) functions with nice structural properties pertinent to games and allowing for efficient optimization by iterative improvement local search style algorithms.

We demonstrate several important combinatorial properties of CLG-functions, allowing for many optimization algorithms, and establish a relation with the subclass of Completely Unimodal (CU) functions, studied by Hammer *et al.* [12], Williamson Hoke [27], and Wiedemann [26]. We also describe a new, compared to our recent [6], subexponential randomized algorithm for CU-functions, CLG-functions, parity, and simple stochastic games, and establish a relation with the class of LP-type problems introduced and investigated by Matoušek, Sharir, and Welzl [22, 19].

Contents

1 Introduction

Over the last decade there has been extensive work on complexity and algorithms for parity games and the related mu-calculus model checking problem, both known to be in NP $\cap$ coNP with open PTIME-membership status. Many of the suggested algorithms are based on the general idea of iterative strategy improvement, although the connections with local search, pseudo-Boolean function optimization, simplex-like algorithms, randomized and combinatorial techniques remain largely unexplored.

The specific technicalities of the games obscure essential structural properties. The major question motivating us was: '*what is an abstract combinatorial structure underlying parity games?*'. Our hope is that a clear succinct definition of this structure will facilitate development and analysis of algorithms for games, and will allow to apply tools from other areas of combinatorial optimization.

There were several premises for our research in this area.

A. The theory of Completely Unimodal (CU-) pseudo-Boolean functions was a starting point, motivation, and important tool of the present work. CU-functions were introduced by Hammer *et al.* [12] and further investigated by Wiedemann [26] and Williamson Hoke [27]; see also Tovey [24] and our Section 2 for a survey. This is a nontrivial class of functions possessing nice structural properties (resembling games) and presumably allowing for efficient optimization[1]. We provided additional evidence to this conjecture by demonstrating that an optimum of a d-dimensional CU function can be found in expected time:

1. $O(2^{0.46d})$ [5]; this was the first bound below $2^{d/2}$, answering the question raised in [26].

2. $O(2^{\sqrt{d}})$ [5, 2]; note that there were no previously known subexponential CU-optimization algorithms [24, 7].

Unfortunately, CU-functions are not completely adequate for the game-theoretic purposes because of the following three drawbacks:

[1] Williamson Hoke [27] conjectures CU-functions can be optimized in polynomial time; actually it is true for the subclass of fully absorbing CU-functions

1. CU-functions are *injective*, but it is cumbersome and unnecessary to evaluate game strategies injectively.

2. CU-functions are *binary*, every coordinate ranges over the Boolean domain. Games usually have non-binary choices, and reducing them to binary games increases the number of vertices quadratically. This eats up the advantages of subexponential algorithms, rendering them exponential.

3. CU-functions have *linearly ordered codomains*, like $\mathbb{N}$ or $\mathbb{R}$. This is a serious restriction requiring comparability of any two game strategies.

The theory of Completely Local-Global functions we develop in this paper is free of all these drawbacks: it allows for coinciding or incomparable values, and non-binary choices. Most importantly, we can still optimize CLG-functions in subexponential time.

B. On the other hand, we were inspired by Ludwig's [17] subexponential algorithm for binary simple stochastic games. We were able [21] to adapt this algorithm to binary parity games, by replacing its linear programming component with graph optimization. But this result was not completely satisfactory for the two reasons: 1) our algorithm involved high-precision arithmetic computations, and 2) it does not extend (remaining subexponential) to non-binary parity games.

C. In a major breakthrough Kalai [16] suggested the first strongly subexponential randomized algorithm for linear programming, followed by other subexponential algorithms by Matoušek, Sharir, Welzl [19, 20]. Building on Kalai's ideas, we suggested a subexponential algorithm for parity games [6] (also extendable to simple stochastic games), without a binary limitation as in [17, 21]. The resulting algorithm [6] and its analysis are reasonably simple. However, the algorithm involves a rather complicated and intuitively inefficient recursive random sampling. In this paper we suggest a new randomized subexponential algorithm for CLG-functions, parity and simple stochastic games. It follows the ideas of Matoušek, Sharir, Welzl [19, 20], is much easier, and avoids this drawback.

D. Local search is a widely investigated general approach for solving hard combinatorial optimization problems. Our CLG-functions, along with CU-functions, simple stochastic games [13, 8, 9, 18], and parity games [25], allow for standard local search-style algorithms, proceeding to one of the better neighbors of a current iterate. Moreover, they can be optimized by *multiple switching algorithms* (see Section 5) allowing for simultaneously changing several parameters and jumping between non-neighbors, an intuitively very appealing idea. Multiple switching for games was investigated in [13, 8, 9, 25], but to our knowledge, none of them suggested any improvements over the trivial 2^d bound (d the number of vertices) for the local search (single or multiple switch). Mansour and Singh [18] improved it to $O(2^{0.772d})$, which is still a far cry from an earlier tight lower and upper bound $\Theta(2^{d/2})$ of Aldous [1], proved for the more general (less structured compared with CU) class of LG-functions (for which every local optimum is global). As mentioned above, our $O(2^{0.46d})$ is the first result showing that a variant of random multiple switch local search beats $\Omega(2^{d/2})$ for CU- and CLG-functions, as well as for binary parity and simple stochastic games.

E. It is worth mentioning here that Tovey [23] introduced and investigated different random distributions on LG-functions and proved low-degree polynomial average case upper bounds under several natural such distributions, for both Greedy Single and Random Single Switch Algorithms (which he calls Optimal and Better adjacency algorithms, respectively). Although his analysis does not directly transfer to 'random' distributions of CU- or CLG-functions (much more difficult to define), it gives at least an indirect evidence in favor that our Single and Multiple Switch algorithms should also be average case polynomial under 'natural' distributions. This has been experimentally confirmed by our extensive simulations using different methods of generating CU-functions [2, 5, 3].

F. Johnson, Papadimitriou, and Yannakakis [14, 28] introduced the complexity class PLS (Polynomial Time Local Search), lying somewhere between PTIME and NP, and identified a lot of well-known problems (including MAX-CUT, MAX2SAT, MAX3SAT under the Flip neighborhood, TSP under k-Opt and Kernighan-Lin's neighborhoods) as being PLS-complete. As usual with the completeness theory, PLS-complete problems are either all polynomial or all superpolynomial. Yannakakis [28] raised a question whether simple stochastic games are PLS-complete. A similar question is appropriate for parity games. It is a simple consequence of our theory (Section 8) that Completely LG functions (and henceforth parity and simple stochastic games) are not PLS-complete, which gives new evidence in favor of the PTIME conjecture.

G. We owe a technical tool of discretizing our previous randomized subexponential algorithm for parity games [6] to Vöge–Jurdziński's [25] discrete strategy evaluation measure. However, we defined a more tight version of this measure allowing for better complexity bounds.

Outline. The paper is organized as follows. In Section 2 we briefly survey main results concerning Completely Unimodal functions. Based on this Section 3 introduces Completely Local-Global (CLG) functions, and Section 4 generalizes their domain from Boolean hypercubes to Cartesian products of finite sets. Section 5 reviews several single and multiple switching algorithms appropriate for CLG-optimization. Section 6 presents a new randomized subexponential algorithm applicable for CU- and CLG-optimization, solving parity and simple stochastic games. Section 7 presents several properties of CLG-functions. Section 8 is devoted to applications of our CLG-functions theory to parity and simple stochastic games. In Section 9 we reduce CLG- to CU-functions, allowing us to transfer many useful results from CU- to a more general class of CLG-functions. In Section 10 we reduce CLG-optimization to LP-type problems. This allows us to reuse any appropriate algorithms for CLG-optimization to solve games. Because of space limitations most proofs are either sketched or moved to the Appendix.

2 Completely Unimodal Functions

To motivate and prepare our definition of Completely Local-Global functions it is helpful to first understand the main results about Completely Unimodal functions, which we briefly survey here. Readers familiar with this material may proceed to Section 3 on CLGFs, and return back by need. More details can be found in [12, 27, 24, 26].

Notational Conventions. Throughout the paper $\mathcal{H}$ denotes the Boolean hypercube $\{0,1\}^d$ (for some $d \in \mathbb{N}$), and $\mathcal{D}$ denotes some partially ordered set. □

Recall that two vertices $x, y \in \mathcal{H}$ are called *neighbors* if they differ in exactly one coordinate (Hamming distance one). A *local maximum* of a function $f : \mathcal{H} \to \mathcal{D}$ is a vertex in $\mathcal{H}$ with value bigger than or equal to all its neighbors. A *global maximum* is a vertex with a function value bigger than or equal to the values of all other vertices. In particular, any global maximum is comparable with every vertex. Local and global minima are defined symmetrically. A k-dimensional *subcube* or *k-face* of $\mathcal{H}$ is obtained if we fix all but k coordinates to either 0 or 1, and allow the remaining k coordinates to vary. The subcube *spanned* by a set of vertices is the smallest subcube containing the set.

Definition 2.1 (Complete Unimodality) *An injective function $f : \mathcal{H} \to \mathcal{D}$ for which all neighbors have comparable values is called* Completely Unimodal *if:*

1. *f has a unique local maximum on every subcube,*

2. *f has a unique local minimum on every subcube.* □

Note that the standard definition of complete unimodality [12, 27, 24, 26] assumes $\mathcal{D}$ *totally* ordered, or even $\mathcal{D}$ equal $\mathbb{R}$ or $\mathbb{N}$. However, the most essential properties of CU-functions rely only on comparability of neighbors on $\mathcal{H}$. This is also more natural in the game-theoretic context, since the strategy measures considered in Section 8 are only partially ordered.

The following remarkable theorem provides equivalent definitions of CU-functions and is fundamental for our studies; cf., Theorems 3.2 and 4.4.

Theorem 2.2 (Williamson Hoke [27]) *For an injective function $f : \mathcal{H} \to \mathcal{D}$ with comparable neighbors the following four statements are equivalent:*

1. *f has a unique local maximum on every subcube,*
2. *f has a unique local minimum on every subcube,*
3. *f has a unique local maximum on every 2-face,*
4. *f has a unique local minimum on every 2-face.* □

Intuitively, the VID-function of the next definition 'points towards better neighbors' (VID stands for 'vector of improving directions').

Definition 2.3 (VID-function) *Let $f : \mathcal{H} \to \mathcal{D}$ be a function such that neighbors have comparable values. The* VID-function *$VID_f : \mathcal{H} \to \mathcal{H}$ of f is defined as follows.*

$$VID_f(x) = (y_1, \dots, y_d),$$

where $y_i = 1$ iff $f(x) < f(x$ with coordinate i flipped$)$. □

The nice property below is due to Williamson Hoke [27].

Theorem 2.4 (Injectivity of VID-function) *VID_f is injective (equivalently, bijective) for any CU-function f.* □

From the point of view of optimization algorithms, CU-functions are particularly interesting because they allow an iterative improvement algorithm to jump from a current vertex v of $\mathcal{H}$ to the next by flipping *simultaneously* all or some coordinates where $VID_f(x)$ is 1. It is guaranteed that the new vertex has a better f-value, because every point is a global minimum in the subcube spanned by its improving directions. Such *multi-switching* algorithms are discussed in detail in the context of the more general CLGFs in Section 5. Combined with random sampling, the random multiple switch algorithm optimizes a d-dimensional CU-function or CLGF in $O(2^{0.46d})$.

Another attractive property of CU-functions, also shared by CLG-functions and games, is that they satisfy the *Hirsh conjecture*.

Proposition 2.5 *Let f be a CU-function. For any point $x \in \mathcal{H}$ at least one of the directions in $VID_f(x)$ points to the maximum. Thus from every point there is a short improving path to the maximum.*

Proof. Consider the subcube F of $\mathcal{H}$ spanned by x and the maximum x^*. If none of the improving directions in x points towards x^*, then F contains two local maxima, a contradiction. $\square$

3 Completely Local-Global Functions

In this section we define a class of functions similar to CU-functions, but more general. These functions share almost all important properties of CU-functions and are more coherent with the applications to games. We show in Section 8 that solving parity and simple stochastic games reduces to finding an optimum on a CLGF, and in Section 9 prove that any CLGF can be transformed to a CU-function.

Definition 3.1 (CLGF) *A function $f : \mathcal{H} \to \mathcal{D}$ with comparable neighbors is* Completely Local-Global (CLG) *if the following conditions hold on every subcube of $\mathcal{H}$:*

1. *any local maximum of f is also global;*
2. *any local minimum of f is also global;*
3. *any two local maxima are connected by a path of local maxima;*
4. *any two local minima are connected by a path of local minima.* □

Note that we do not stipulate injectivity. CLGFs differ from LG-functions [1, 24], and can be optimized in sub-$2^{d/2}$ time (the lower bound for LG-functions.)

Proposition 2.5 holds for CLG-functions by the same argument. The following theorem is analogous to Theorem 2.2 for CU-functions. It localizes and simplifies the proof that a function is CLG.

Theorem 3.2 *Let $f : \mathcal{H} \to \mathcal{D}$ be a function with comparable neighbors such that its restriction to any 2-dimensional subcube of $\mathcal{H}$ satisfies conditions 1, 2, and either 3 or 4 in Definition 3.1. Then f is CLG.*

Proof. We first note that on a 2-dimensional subcube, a function that fulfills conditions 1, 2, and either 3 or 4 fulfills all the conditions. In fact, a 2-face has two local maxima not connected by a path of local maxima, if and only if it has two local minima with no path of local minima between.

We prove that f is CLG on every subcube F of $\mathcal{H}$ by induction on the dimension d_F of F. If $d_F \leq 2$ the theorem is immediate from the definition. Assume it holds for all cubes of dimension $< n$ and let $d_F = n$. Suppose first there is either 1) a local maximum that is not global, or 2) two local maxima not connected by a path. In the first case, the local maximum a is global on every $n-1$-dimensional face of F by induction hypothesis. It follows that it has value greater or equal to all vertices on F except the one on the long diagonal (with Hamming distance n), which we denote by b. Note that b has value strictly greater than all its neighbors: otherwise it would be smaller or equal to the value of some neighbor, which in turn would be smaller or equal to the value of a, so a would be a global maximum.

In the case of two local maxima not connected by a path, the situation is identical: two vertices a and b with Hamming distance n.

In both cases, consider any neighbor c of b and the $n-1$-face G spanned by c and a. No other vertex can be a local maximum on G, since by the inductive hypothesis it would have value identical to both a and b and connect them with a path, contradicting either assumption 1) or 2). Since a is the only local maximum on G, there must be a strictly increasing path from c to a on F. Symmetrically, there must be an increasing path from every neighbor of a to b. This contradiction concludes the proof.

The proof for minima is symmetric. $\square$

The motivation for defining CLGFs is two-fold. On one hand, it captures the structure of the strategy measures for parity and simple stochastic games in a succinct way (see Section 8). On the other hand, it includes and reduces to the well-studied CU-functions (see Section 9). As it turns out, CLGFs and CU-functions are very similar; they share almost all important properties, and allow for both multiple switching and subexponential algorithms.

Figure 1 shows a 2-dimensional function for which all local maxima and minima are global; however, they are not connected by paths of maxima (minima), so the function is not CLG. Ruling out such cases is important for showing the property of multi-switching, discussed in Section 5.

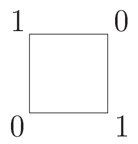

Figure 1: A two-dimensional hypercube where multi-switching does not work

4 General (Non-Binary) Completely Local-Global Functions

In this section we generalize CLG-functions by extending the domain to a product of arbitrary finite sets, rather than a power of $\{0, 1\}$. This reflects games more naturally, since games usually allow for non-binary choices. Reducing non-binary to binary parity games gives a quadratic increase in the number of vertices (hence the dimension of the corresponding hypercube), which decreases the efficiency of the algorithms. Defining algorithms directly on generalized structures improves efficiency.

Definition 4.1 (Hyperstructure) *For each $j \in \{1, \ldots, d\}$ let $\mathcal{P}_j = \{e_{j,1}, \ldots, e_{j,\delta_j}\}$ be a finite set. Call $\mathcal{P} = \prod_{j=1}^{d} \mathcal{P}_j$ a* hyperstructure, *or* structure *for short.* □

A *substructure* $\mathcal{P}' \subseteq \mathcal{P}$ is a product $\mathcal{P}' = \prod_{j=1}^{d} \mathcal{P}'_j$, where $\mathcal{P}'_j \subseteq \mathcal{P}_j$ for all j. If each $\mathcal{P}'_j$ has only two elements, then identify $\mathcal{P}'$ with $\mathcal{H}$. Call $\mathcal{P}'$ a *facet* of $\mathcal{P}$ if there is a j such that $\mathcal{P}'_k = \mathcal{P}_k$ for all $k \neq j$, and $\mathcal{P}'_j$ has only one element. Say that $x, y \in \mathcal{P}$, are *neighbors* iff they differ in only one coordinate. Thus each $x \in \mathcal{P}$ has exactly $\sum_{j=1}^{d}(\delta_j - 1)$ neighbors. The neighbor relation defines a graph with elements of the hyperstructure as nodes, allowing us to talk about paths and distances in the hyperstructure. A structure $\mathcal{P} = \prod_{j=1}^{d} \mathcal{P}_j$ has d dimensions and $n = \sum_{j=1}^{d} |\mathcal{P}_j|$ facets.

Definition 4.2 (VID-function) *Let $f : \mathcal{P} \to \mathcal{D}$ be a function on $\mathcal{P} = \prod_{j=1}^{d} \mathcal{P}_j$ with comparable neighbors. Its corresponding* VID-function *VID_f is a function from $\mathcal{P}$ to $\prod_{j=1}^{d} 2^{\mathcal{P}_j}$, defined in the following way. Let $v \in \mathcal{P}$ and let u be the vertex reached from v by switching to $e_{j,i}$ in coordinate j. The value $VID_f(v)$ contains $e_{j,i}$ in coordinate j iff $f(v) < f(u)$.* □

The notion of CLG-function extends to these more general structures, as follows. A *local maximum* of a function $f : \mathcal{P} \to \mathcal{D}$ is a vertex in $\mathcal{P}$ with value bigger than or equal to all its neighbors. A *global maximum* is a vertex with a function value bigger than or equal to the values of all other vertices. In particular, any global maximum is comparable with all other vertices. Local and global minima are defined symmetrically.

Definition 4.3 (CLG-function on a hyperstructure) *Let $f : \mathcal{P} \to \mathcal{D}$ be a function such that all neighbors have comparable values. Say that f is a* CLG-function *if the following properties hold on every substructure of $\mathcal{P}$:*

1. *any local maximum of f is also global;*
2. *any local minimum of f is also global;*
3. *any two local maxima are connected by a path of local maxima;*
4. *any two local minima are connected by a path of local minima.*

By a CLG-structure *we mean a CLG-function together with the underlying hyperstructure; $w(\mathcal{P})$ denotes the maximum value of f on $\mathcal{P}$.* □

Theorem 3.2 generalizes as follows. Denote by *binary 2-face* any substructure $\mathcal{P}'$ of $\mathcal{P}$ isomorphic to $\{0,1\}^2$, that is, any substructure with $|\mathcal{P}'_j| = 1$ for all values of j except two, $j \in \{j_0, j_1\}$, with $|\mathcal{P}'_{j_0}| = |\mathcal{P}'_{j_1}| = 2$.

Theorem 4.4 *Let $f : \mathcal{P} \to \mathcal{D}$ be a function with comparable neighbors. If the restriction of f to any binary 2-face fulfills conditions 1, 2 and either 3 or 4 of Definition 3.1, then f is CLG.*

Proof. Consider any two local maxima a and b on $\mathcal{P}$. There is a unique minimal hypercube $\mathcal{H}$ that contains a and b and is a substructure of $\mathcal{P}$. By Theorem 3.2, the restriction of f to $\mathcal{H}$ is CLG and thus a and b have the same value and are connected by a path of local maxima. The reasoning for local minima is symmetrical. □

The concept of this proof, looking at substructures that are hypercubes and applying theorems for CLG-functions on hypercubes, can be used in many proofs for the general structures.

Like CLGFs defined on hypercubes, CLGFs on hyperstructures allow for multi-switching algorithms, as described in the next section.

5 Switching Algorithms

In this section we survey iterative improvement algorithms for optimizing CLGFs, first on hypercubes and then on hyperstructures. All algorithms are monotone, proceeding to better vertices until they reach the optimum. The most known and used such algorithms are greedy (optimal adjacency) and random (better adjacency) local search. The random version is applicable to CLGFs, and the conjecture from [27], that it optimizes CU-functions in polynomial time, might well be extended to CLGFs on hypercubes. The greedy version, however, is not applicable in our setting, since we only assume the codomains of our functions to be partially ordered. Two neighbors may be better than the current vertex, but incomparable to each other.

The following proposition suggests that an iterative improvement algorithm may jump simultaneously in the directions of several better neighbors, allowing for more 'aggressive' improvement steps.

Proposition 5.1 (Multi-switching) *Let $f : \mathcal{H} \to \mathcal{D}$ be a CLGF and $x \in \mathcal{H}$. Let $y \in \mathcal{H}$ be a vector obtained by flipping some coordinates in x corresponding to ones in $VID_f(x)$. Then $f(y) > f(x)$.*

Proof. Consider a vertex $x \in \mathcal{H}$. On the subcube F defined by x's improving directions, it must be the unique local minimum. Indeed, if there was another local minimum, there would be a path on F from it to x containing only local minima by Definition 3.1. Thus one of the neighbors of x on F would have the same value as x, contradicting the fact that on F all directions from x are improving. □

In particular the following two iterative improvement algorithms can be used to optimize CLGFs on $\mathcal{H}$.

All Profitable Switches Algorithm (APSA) iteratively proceeds from a current iterate v_0 by simultaneously switching *all* coordinates corresponding to better neighbors of v_0 (i.e., jumping along the big diagonal in the subcube generated by all improving directions). In the context of simple stochastic games this algorithm is known as Hoffman–Karp's [13, 9]. In the context of parity games this is Vöge–Jurdziński's algorithm [25].

Random Multiple Switches Algorithm (RMSA) iteratively proceeds from a current iterate v_0 by simultaneously switching in *some* coordinates corresponding to better neighbors of v_0, each improving direction being chosen with probability 1/2 (i.e., jumping to some uniformly and randomly chosen vertex in the subcube generated by all improving directions).

We settled the following nontrivial upper bounds on the expected number of iterations for APSA and RMSA on CU-functions in [5]: $O(2^{0.775d})$ for RMSA, $O(2^{0.46d})$ for RMSA with random sampling (start from the best of $2^{0.46d}$ randomly chosen starting points) and $O(2^{d-(1-\varepsilon)\log d})$ (for any $\varepsilon > 0$) for APSA. Similar (except $2^{0.46d}$) bounds were established in [18] for Markov decision processes. These bounds are not known to hold on CLGFs. In fact, the analysis in [5] relies on Theorem 2.4 which does not hold on CLGFs.

Ludwig's algorithm. In [2, 5] we show how Ludwig's subexponential algorithm for optimizing simple stochastic games can be used to optimize CU-functions. The argument straightforwardly generalizes to CLGFs. Thus it can be used to optimize any CLGF $f : \mathcal{H} \to \mathcal{D}$ in expected time $2^{O(\sqrt{d})}$. Unfortunately, it only applies to hypercubes, but see Section 6 for an appropriate extension to hyperstructures.

Theorem 5.1 generalizes to CLGFs defined on hyperstructures, as follows.

Theorem 5.2 (Multi-switching in hyperstructures) *Let $f : \mathcal{P} \to \mathcal{D}$ be a CLGF, $x \in \mathcal{P}$, and $y \in \mathcal{P}$ be a vector such that $y_j \neq x_j$ implies $f(x) < f(x$ with coordinate j changed to $y_j)$. Then $f(x) < f(y)$.*

Proof. The substructure spanned by x and y is a hypercube, and all directions on this hypercube are improving for x. The claim now follows from Theorem 5.1. □

Hyperstructures give more options than hypercubes for defining multi-switching algorithms, since there may be many improving directions in each coordinate. Since two better neighbors are not necessarily comparable, an extension of All Profitable Switches for hyperstructures chooses a *random* better neighbor in all coordinates where there is a better neighbor. An extension of Random Multiple Switches chooses in each coordinate uniformly at random among better neighbors *and* the current value. In particular applications, such as parity games, it might be possible to use domain-specific heuristics to choose among better neighbors.

6 Subexponential Randomized Optimization

In [6] we presented a randomized subexponential algorithm for parity games, based on the ideas of Kalai's algorithm for linear programming [16]. This algorithm can be easily adapted to optimize CLGFs. It is, however, rather complicated and involves some intuitively inefficient random sampling. Here we show how the LP-algorithm by Matoušek, Sharir, and Welzl [19, 20] can be modified for CLGFs, yielding a simpler and possibly more efficient subexponential randomized algorithm for the problem (and thus also for parity games). It has expected running time at most $2^{O(\sqrt{d \log(n/\sqrt{d})} + \log n)}$, where $n = \sum_{j=1}^{d} |\delta_j|$ is the number of facets of $\mathcal{P}$.

Roughly, the basic concept is to identify facets of the CLG-structure $\mathcal{P}$ to be optimized with constraints in the linear programming setting, see also Section 10. A set G of facets defines a substructure consisting of all vertices that do not belong to any facet outside G. The best vertex in this structure translates into a basis for a set of constraints in linear programming.

Algorithm 1: MSW-Style Optimization Algorithm
Input: $\mathcal{P}$ is a CLG-structure, v is a vertex on $\mathcal{P}$
Output: Optimal vertex on $\mathcal{P}$
Optimize($\mathcal{P}, v$)
(1) **if** $\mathcal{P} = v$
(2)   **return** v
(3) **else**
(4)   choose a random facet F of $\mathcal{P}$, not containing v
(5) $v' \leftarrow$ Optimize($\mathcal{P} \setminus F, v$)
(6)   **if** neighbor v'' of v' on F is better than v'
(7)   **return** Optimize(F, v'')
(8)   **else return** v'

Note that this algorithm generalizes Ludwig's algorithm, discussed in Section 5. If $\mathcal{P}$ is a hypercube, the two algorithms are the same.

The algorithm is correct since any vertex without better neighbors in a CLG-structure is globally optimal. It always terminates since the recursive calls in lines (5) and (7) are made on strictly smaller substructures.

It remains to show that the bound from [20] holds in this setting. This is done by translating into the concepts of [20], thus showing that the same recurrence for the running time holds. The following definition will be useful.

Definition 6.1 (Extreme facet) *A facet F of a CLG-structure $\mathcal{P}$ is* extreme *if it contains all local maxima on $\mathcal{P}$.* □

The second recursive call on line (7) of the algorithm will be performed only when F is extreme on $\mathcal{P}$. At most one facet in each coordinate can contain all maxima, and thus there can be at most d extreme facets. This means that the second call will be made with probability at most $\frac{d}{n-d}$. If the current vertex belongs to $d-k$ extreme facets, the probability is at most $\frac{\min(k,n-d)}{n-d}$.

Definition 6.2 (Hidden dimension) *Given a CLG-structure $\mathcal{P}$ and a vertex $v \in \mathcal{P}$, the* hidden dimension *of the pair $(\mathcal{P}, v)$ is d minus the number of facets F of $\mathcal{P}$ containing every $u \in \mathcal{P}$ with $f(u) > f(v)$ or $u = v$.* □

Intuitively, if $(\mathcal{P}, v)$ has hidden dimension k, then v and all vertices with better values than v belong to some k-dimensional substructure $\mathcal{P}'$ of $\mathcal{P}$. In particular, if the hidden dimension is 0, then v is a maximum on $\mathcal{P}$.

The algorithm will never visit a vertex that does not belong to $\mathcal{P}'$. There are $d-k$ facets, $F_1, F_2, \ldots, F_{d-k}$, that contain $\mathcal{P}'$. In the worst case there are k more extreme facets. For each such facet, consider the best value that does not belong to it. Suppose for the moment, that these values are totally ordered. Enumerate the facets so that

$$w(\mathcal{P} \setminus F_{d-k+1}) \leq \cdots \leq w(\mathcal{P} \setminus F_{d-k+i}) \leq \cdots \leq w(\mathcal{P} \setminus F_d).$$

Suppose the algorithm chooses facet F_{d-k+i}. Then $w(v') = w(\mathcal{P} \setminus F_{d-k+i}) < w(v'')$. Every vertex with a better value than v' that belongs to F_{d-k+i} must also belong to F_{d-k+j} for all $1 \leq j < i$. Thus v'' belongs to $(d-k+i)$ facets that contain all vertices with better values, and the hidden dimension of the pair (F_{d-k+i}, v'') for the second call is at most $(k-i)$.

If the best values outside the remaining extreme facets are not totally ordered, this only benefits the algorithm. The values are partially ordered, and if facet F_{d-k+i} is chosen, v'' will belong to all facets that do not have a strictly bigger value for $w(\mathcal{P} \setminus F_j)$, and they will all contain all vertices with better values. This is because if $w(\mathcal{P} \setminus F_{d-k+i})$ and $w(\mathcal{P} \setminus F_j)$ are incomparable, then any value better than $w(\mathcal{P} \setminus F_{d-k+i})$ will be better than or incomparable to $w(\mathcal{P} \setminus F_j)$.

This discussion is enough to get the same recurrences for the numbers of tests on line (6) and jumps to v'' on line (7) as [20] gets for the numbers of violation tests and basis computations, respectively, in the linear programming setting. The recurrences are solved in [20], and from the solutions it is easy to infer that the expected running time of the algorithm on CLG-structures is at most $2^{O(\sqrt{d \log(n/\sqrt{d})}+\log n)}$, as long as function evaluation and comparison of function values can be performed in polynomial time.

Note that the argument above relies only on the first property of CLGFs from Definition 4.3, and not on properties 2, 3, 4. Thus, for the algorithm to work it is enough that neighbors have comparable values, all local maxima are global, and that the same holds on all substructures. It is also possible to reduce the problem of optimizing CLGFs to what [19] calls an *LP-type problem*, and then directly apply the algorithm presented in that paper; see Section 10.

7 Important Properties of CLGFs

This section shows a number of strong properties of CLGFs, confirming that the notion is useful, providing a deeper insight in their structure, and helpful in other proofs. All theorems, except Theorem 7.1 and its corollary, were proved first for CU-functions.

Theorem 7.1 (Subcubes with uniform values) *Let $f : \mathcal{H} \to \mathcal{D}$ be a CLGF, and let p be a path on $\mathcal{H}$ such that all vertices on p have the same value. Then all vertices on the smallest subcube containing p have the same value.*

Proof. By induction on the length of p. Assume the theorem holds for all paths shorter than n and let p have length n. Consider the path consisting of all but the last vertex a of p. By inductive assumption, all vertices on the smallest face F containing it have the same value. If $a \in F$ then we are done; assume $a \notin F$ and let k be the dimension of F. From every vertex on F with Hamming distance k to a there is a path to a of length k on which all vertices have the same value. Thus all vertices on faces spanned by these vertices and a have the same value. Together with F, these faces contain all but one vertex of the $k+1$-dimensional face G containing both F and a. The last vertex in G must also have the same value since it shares a 2-dimensional face with three vertices of the same value. □

Corollary 7.2 *Any CLGF on $\mathcal{H}$ has 2^m global maxima (minima), for some $m \leq d$, all located on the same m-face.* □

The next theorem was shown in [26] for CU-functions.

Theorem 7.3 *Let $f : \mathcal{P} \to \mathcal{D}$ be a CLGF, $v \in \mathcal{P}$, and d the dimension of $\mathcal{P}$. Then there is a path of vertices with strictly increasing (decreasing) f-values from v to a global maximum (minimum) of length at most d.*

Proof. We will prove a slightly stronger claim, namely that there is a path to the *nearest* maximum v^*, in the sense of the Hamming distance $h = hd(v, v^*)$, and that its length is exactly h.

First note that Proposition 2.5 generalizes without difficulty to CLGFs, using the proof technique of Theorem 4.4. Thus for a non-maximal vertex v, at least one of its better neighbors u has shorter Hamming distance to the nearest maximum v^*. The theorem is now proved by induction on the Hamming distance from v to v^*. Suppose it holds for all shorter distances. Then there is a path from u to v^* with length $hd(u, v^*)$. Since the step from v to u is improving there is an improving path from v to v^* with length $hd(v, v^*)$. The proof for minima is symmetrical. Now it is sufficient to note that v and v^* always share a substructure $\mathcal{H}$ of $\mathcal{P}$ that is a hypercube. □

The following theorem was mentioned without a proof for CU-functions in [26], and it also holds for CLGFs.

Theorem 7.4 (Projection) *Let $f : \mathcal{H} \to \mathcal{D}$ be a CLGF and $k \in \{0, \ldots, d\}$. Define the* projected *$(d-k)$-variable function $\widetilde{f}$ as follows.*

$$\widetilde{f}(x) = \max\{f(x, y) : y \in \{0, 1\}^k\}$$

Then $\widetilde{f}$ is a CLGF. Moreover, any maximum of $\widetilde{f}$ is also a maximum of f.

(Of course, it is not necessary to minimize over the last coordinates, it may be any k coordinates. The theorem is also true, with symmetric proof, if we substitute 'min' for 'max'.)

Proof. We only prove the case $k = 1$. For bigger k, iterate the one-coordinate projection k times. By Theorem 3.2 we need to prove that neighbors are comparable and for every 2-face that every local maximum or minimum is global, and local maxima are connected by paths.

Let $\widetilde{\mathcal{H}} = \{0, 1\}^{d-1}$ and let $\widetilde{x}, \widetilde{y} \in \widetilde{\mathcal{H}}$ be neighbors. By the definition there are $x, y \in \mathcal{H}$, both on the same 2-face F of $\mathcal{H}$, such that $f(x) = \widetilde{f}(\widetilde{x})$ and $f(y) = \widetilde{f}(\widetilde{y})$. One of x and y is the global maximum on F, so its f-value is comparable with all other f-values on F, in particular $\widetilde{f}(\widetilde{x})$ and $\widetilde{f}(\widetilde{y})$ are comparable.

Assume some 2-dimensional face $\widetilde{F}$ of $\widetilde{\mathcal{H}}$ has a local maximum that is not global, a local minimum that is not global, or two local maxima that are not connected by a path of local maxima. In all three cases, it follows that $\widetilde{F}$ has at least two local maxima $\widetilde{x}, \widetilde{y}$ on a diagonal as in Figure 2.

Let $X, Y \subset F$ be the 1-faces over which the maxima are taken when computing $\widetilde{f}(\widetilde{x})$ and $\widetilde{f}(\widetilde{y})$, respectively. Let $x \in X$ and $y \in Y$ be vertices where the maxima are attained. If both x and y are global maxima, then by Theorem 7.1 all vertices on the face spanned by x and y have the same value. But this face contains vertices over which the maxima are taken when

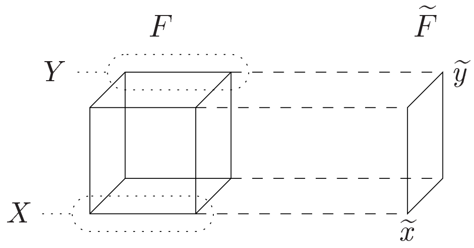

Figure 2: $\widetilde{x}$ and $\widetilde{y}$ are local maxima of $\widetilde{f}$ on $\widetilde{F}$

computing $\widetilde{f}(\widetilde{z})$ for $\widetilde{z} \in \widetilde{F} \setminus \{\widetilde{x}, \widetilde{y}\}$. It follows that all vertices on $\widetilde{F}$ have the same $\widetilde{f}$-values, so $\widetilde{f}$ is CLG. If, on the other hand, x and y have different values, say $x < y$, then y is a global maximum on F and no other vertex of $F \setminus Y$ is a global maximum. By Theorem 7.3, there is an improving path from x to y on F. This path must pass through some vertex over which the maximum is taken when computing $\widetilde{f}(\widetilde{z})$ for some $\widetilde{z} \in \widetilde{F} \setminus \{\widetilde{x}, \widetilde{y}\}$. Since the path is improving, $\widetilde{f}(\widetilde{z}) > \widetilde{f}(\widetilde{x})$, contradicting the assumption that $\widetilde{x}$ is a local maximum. □

Note that the previous theorem allows for a different kind of optimization algorithm than the iterative improvement algorithms described in Section 5 (as noted also in [26]). Rather than optimizing f directly, we optimize $\widetilde{f}$ using any known algorithm on the $d-k$-cube. The $\widetilde{f}$-value of a vertex of $\widetilde{f}$ is computed using any known optimization algorithm on the k-cube. To our knowledge this approach is rather unexplored.

Theorem 7.5 (for CU-functions) was first proved in [27].

Theorem 7.5 *For any CLGF $f : \mathcal{H} \to \mathcal{D}$ and any two vertices u and v of $\mathcal{H}$, if $VID_f(v) \subset VID_f(u)$, then $f(v) > f(u)$.*

Proof. Let k be the number of ones in $VID_f(v)$. Consider the vertex a that coincides with v in all coordinates where $VID_f(v)$ has ones, and with u in the other coordinates. Thus a and v share a $(d-k)$-face on which v has no improving directions. So v is the a maximum on this subcube, and $f(a) \leq f(v)$. Furthermore, a and u share a k-face on which u has k improving directions and is the unique global minimum. Thus $f(u) < f(a) \leq f(v)$. □

Proposition 2.5 and Theorems 7.1, 7.4, and 7.5 generalize without difficulties to CLGFs defined on hyperstructures, using the proof technique of Theorem 4.4.

8 Application to Games

In this section we demonstrate, as was promised in the introduction, that CLG-functions capture the combinatorial structure of strategy evaluation functions of parity and simple stochastic games. First, we very briefly sketch the key definitions concerning games.

Definition 8.1 *A* parity game *is an infinite game played on a finite directed leafless graph $G = (V_0 \uplus V_1, E)$ with vertices assigned integer colors. Players 0 and 1 move a pebble along edges of the graph, starting in an initial vertex. Player i makes moves (by choosing a successor) from vertices in V_i. Player* 0 *wins if the highest color encountered infinitely often is even, while Player* 1 *wins if it is odd.* □

For more information on parity games see [10, 25, 29, 6].

Definition 8.2 *A* simple stochastic game *is an infinite game played on a directed graph $G = (V_{max} \uplus V_{min} \uplus V_{avg} \uplus \{v_0, v_1\}, E)$. The vertices v_0 and v_1 are called* sinks. *All other vertices have two outgoing edges. Two players, Max and Min, move a pebble along edges of the graph. Max and Min choose the moves in their vertices V_{max} and V_{min} respectively. The moves from V_{avg} are decided by tossing a fair coin. The game ends when the pebble reaches a sink. Max aims at maximizing and Min at minimizing the probability of reaching v_1.* □

For more information on simple stochastic games, see [8, 9, 17]. One can define parity and simple stochastic games of arbitrary outdegree, but the reduction of such games to binary ones makes the algorithm from [17] exponential, while our algorithms from [6] and Section 6 are subexponential in that case.

A (positional) *strategy* for a player in a parity or simple stochastic game is a set of edges containing exactly one edge leaving each vertex of the player. Both games are known to be determined and solvable in such strategies; see, e.g., [8, 10, 29, 4].

8.1 Reducing Parity Games to CLGFs

A common approach to solving such games, i.e., finding an optimal strategy, is to assign values to strategies in a way that optimal strategies have the best values, and then search the strategy space guided by the value assignments [13, 8, 9, 25, 6]. In this section we show that using an appropriate value assignment, such as that by Vöge and Jurdziński [25], or our measure from [6], yields a CLG-function. The same holds for simple stochastic games if each strategy σ of Max is assigned a value equal to the sum, over all vertices v, of probabilities reaching v_1 from v, when Max uses σ and Min plays optimally against σ.

For each vertex v in a parity game, let δ_v be the number of edges leaving v. The space of strategies for Player 0 in a parity game with $|V_0| = d$ and $\sum_{v \in V_0} \delta_v = n$ is a d-dimensional hyperstructure with n facets.

Theorem 8.3 *Let $\mathcal{P}$ be the hyperstructure corresponding to the strategy space of a parity game G, and let $f_G : \mathcal{P} \to \mathcal{D}$ be the assignment of values to strategies given by the evaluation from [25] or [6]. Then f_G is CLG.*

Proof. By Theorem 4.4, it is enough to show that f_G is CLG on all binary two-dimensional substructures of $\mathcal{P}$. This is the same as saying that for any restriction of G to a game G_2 in which Player 0 has choices in only two vertices, and has exactly two choices in each of them, f_G has the following properties on the two-dimensional hypercube $\mathcal{H}$ corresponding to the four possible strategies:

1) all neighbors have comparable values.

2) every local maximum is also global;

3) every local minimum is also global;

4) every pair of local minima are connected by a path of local minima.

Properties 1) and 2) were shown (for any-dimensional faces) in [25] and [6], respectively. We now prove property 3), arguing by symmetry. Let σ_0 and σ_1 be two strategies of Player 0 that are local minima. Let G_2' be the game G_2 except all vertices belong to Player 1, and consider optimizing Player 1's strategy rather than Player 0's. Let τ_0 and τ_1 be the strategies of Player 1 in G_2' that coincide with σ_0 and σ_1 in vertices that were Player 0's in G_2, and coincide with the optimal response strategies of Player 1 to σ_0 and σ_1 in other vertices. Now there are no attractive switches for Player 1: the switches in vertices that were Player 0's are non-attractive since σ_i are assumed to be local minima, and the switches in other vertices are non-attractive since the strategy in those vertices is optimal for Player 1. Thus both τ_0 and τ_1 are optimal (for Player 1), so they have the same value. It now suffices to note that the values of σ_0 and σ_1 in G_2 are identical to the values of τ_0 and τ_1 in G_2', respectively. This shows 3).

It remains to prove 4). Assume, on the contrary, that two strategies σ_0 and σ_1 are minimal but there is no path on $\mathcal{H}$ of strategies with identical values between them. Then they are on the diagonal, have the same values, and the other strategies have bigger values. For the evaluation from [25] where multi-switching is known to work this is impossible, since from σ_0 there would be two attractive switches, but making them both simultaneously, jumping to σ_1, would not give an improvement. The proof for the evaluation from [6] (also valid for the one from [25]) is more involved and requires understanding of the evaluation function.

Let a and b be the vertices where Player 0 has choices. For $x, y \in \{0, 1\}$, denote by $[x, y]$ the strategy corresponding to corner (x, y) on the two-dimensional face. We may without loss of generality assume that $[0, 0]$ and $[1, 1]$ have minimal values while $[1, 0]$ and $[0, 1]$ have non-minimal values (the case when $[1, 0]$ and $[0, 1]$ are minimal is symmetric). Note that $[0, 1]$ and $[1, 0]$ both are local maxima, so they are also global and have the same value. See Figure 3.

$$\begin{array}{ccc} [0,1] & \succ & [1,1] \\ \curlyvee & & \curlywedge \\ [0,0] & \prec & [1,0] \end{array}$$

Figure 3: $[1, 0]$ and $[0, 1]$ are best, the other strategies are worst

Denote by $val_\sigma(v)$ the value of vertex v under strategy σ. Let a^+, b^+ be the vertex values of a and b under the strategies $[0, 1]$ and $[1, 0]$ and let a^-, b^- be the values under $[0, 0]$ and $[1, 1]$. Since the switch from $[0, 0]$ to $[1, 0]$ improves the value in a we have $a^- \prec a^+$, and since the switch from $[0, 0]$ to $[0, 1]$ improves the value in b we also have $b^- \prec b^+$.

Note that Player 1 in his optimal response strategy against $[0, 0]$ reaches a; if not, the same response strategy would be possible against $[1, 0]$, resulting in $b^+ \preceq b^-$, a contradiction. Symmetric arguments show that b reaches a under strategy $[1, 1]$ and that a reaches b under both strategies $[0, 0]$ and $[1, 1]$. In other words, a and b are on the same loop under strategies $[0, 0]$ and $[1, 1]$.

Let t bet the common loop tint of a^- and b^-. Consider four cases depending on whether t coincides with a or b, or is on the path from a to b or from b to a against strategy $[0, 0]$.

Case 1: $a = t$. Construct a play starting in b against $[0, 1]$ as follows. First follow the same path as against $[1, 1]$ until we reach a. From a, follow the same path as against $[0, 0]$ until we either reach b or interfere with the choices already made. See Figure 4. This gives a loop over t with the same path from b to t as in $[1, 1]$. Thus $val_{[0,1]}(b) \preceq val_{[1,1]}(b)$, contradicting our result that $val_{[0,1]}(b) = b^+ \succ b^- = val_{[1,1]}(b)$.

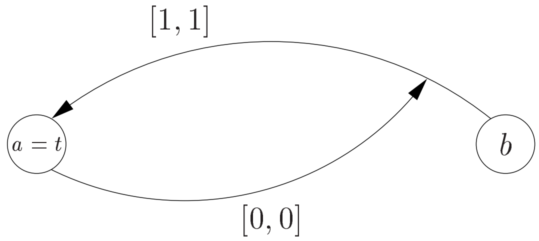

Figure 4: If $a = t$ (Case 1), then Player 1 can achieve the same value in b under strategy $[0, 1]$ as under $[1, 1]$.

Case 2: $b = t$. The proof is symmetric to that of Case 1.

Case 3: t is on the path from a to b under $[0, 0]$. We now construct an auxiliary play against strategy $[0, 1]$ looping over t both from a and from b. First select the same path from a to t as against $[0, 0]$. Then follow the path from b to a as against $[1, 1]$ until either a or another vertex on the constructed path from a to t is reached. Finally extend the path from t to b as against $[0, 0]$, until either b or any other vertex on our constructed path is reached. See Figure 5.

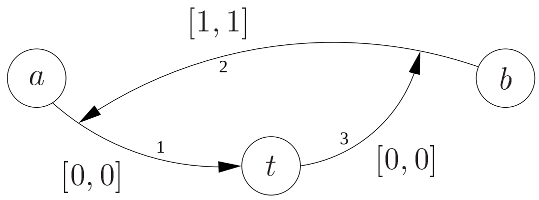

Figure 5: If t is on the path from a to b under $[0, 0]$ (Case 3), then Player 1 can achieve the same value in a under strategy $[0, 1]$ as under $[0, 0]$.

Since the constructed path can be chosen by Player 1 against $[0, 1]$ and it gives the value a^- in a, we must have $a^+ = val_{[0,1]}(a) \preceq val_{[0,0]}(a) = a^-$. But this contradicts our earlier conclusion that $a^+ \succ a^-$.

Case 4: t **is on the path from** b **to** a **under** $[0, 0]$**.** The proof is symmetric to case 3.

This concludes the proof. □

An analog of Theorem 8.3 also holds for simple stochastic games. The proof has the same structure as for parity games, relies on basic properties of simple stochastic games shown in [8] and [13], and will appear in the full paper.

8.2 Non-PLS-Completeness of Parity and Simple Stochastic Games

As mentioned in the introduction, the complexity class PLS (Polynomial Time Local Search) [14, 28] intuitively captures local search problems where the next feasible solution is taken from a polynomially big neighborhood. It is immediate that CLG optimization using a single-switching algorithm is in PLS. Yannakakis [28] showed PLS-completeness of many natural problems and asked whether simple stochastic games are PLS-complete. Luckily, the following theorem gives the negative answer (also for other problems we consider). This is good news, since tight PLS-complete problems should have exponential lower bounds on improving paths to the optimum for any pivoting rules, including randomized and the rules that may not be implementable in polynomial time.

Theorem 8.4 *The problems of optimizing CU-, CLG-functions, as well as finding optimal strategies in parity and simple stochastic games (all with the standard 1-flip neighborhood structure) are not complete with respect to the tight PLS-reductions.*

Proof. This follows from [28, Lemmas 11, 12, Theorem 13], and our Theorem 7.3, hence by Theorem 8.3 for parity and simple stochastic games. □

8.3 New Randomized Subexponential Algorithm for Parity Games

We now recast the MSW-style randomized subexponential algorithm from Section 6 to work directly on parity games. This new algorithm is *different* and simpler than the one we presented in [6]. In this algorithm *switch*(σ', e)

substitutes e for the edge leaving the same vertex in σ', and 'e improves σ'' means that the strategy $switch(\sigma', e)$ is better than σ'. The details may be understood from [6], and together with the analysis will appear in the full paper. It is not difficult to modify the algorithm for simple stochastic games of arbitrary outdegree. Recall that the algorithm of Ludwig [17] applies to binary games, and becomes *exponential* after the standard reduction of non-binary to binary games; see [6] for details.

Algorithm 2: MSW-Style Parity Game Algorithm
Input: $G = (V, E)$ is a parity game, σ is a strategy in G
Output: Optimal strategy in G

```
SOLVE(G = (V, E), σ)
if σ = E
  return σ
else
  choose a random edge e = (v, u) ∈ E \ σ
  σ' ← SOLVE((V, E \ {e}), σ)
  if e improves σ'
    E' ← E \ {e' : e' ≠ e, e' starts in v}
    return SOLVE((V, E'), switch(σ', e))
  else return σ'
```

In a parity or simple stochastic game with d vertices, there are at most $O(d^2)$ edges, making the expected running time of the algorithm at most $2^{O(\sqrt{d \log d})}$.

8.4 UP ∩ coUP-Membership

Jurdziński [15] showed that parity games is in the complexity class UP ∩ coUP, the subclass of NP ∩ coNP where certificates can be taken to be unique. We mention briefly that the reduction from parity games via CLGFs to CU-functions gives another proof of this fact. Moreover, we get the same result for simple stochastic games. As certificate we take the unique global maximum on the CU-function.

9 Reducing CLGFs to CU-functions

Given the numerous results for CU-functions ([12, 26, 27, 24, 2, 5, 3]), it is interesting and important that CLGFs on hypercubes efficiently reduce to CU-functions. Recall that all theorems in Section 5 and 7 except Theorem 7.1 were first shown for CU-functions, and that the upper bounds for the number of iterations of APSA and RMSA on CU-functions mentioned in Section 5 are not known to hold on CLGFs. By definition, any CU-function is a CLGF. Conversely, it is possible to turn any CLGF on a hypercube into a CU-function by inducing an artificial order on neighbors with equal values, as follows.

Definition 9.1 *Let $f : \mathcal{H} \to \mathcal{D}$ be a CLGF. Define a partial order on $\mathcal{D}_{CU} = \mathcal{D} \times \mathcal{H}$ as follows. For any $(\alpha, x), (\alpha', x') \in \mathcal{D}_{CU}$, say that $(\alpha, x) < (\alpha', x')$ if:*

1. *$\alpha < \alpha'$, or*
2. *$\alpha = \alpha'$ and $x < x'$, where the comparison of coordinates is lexicographic.*

It is easily verified that this is a partial order.

Theorem 9.2 *If $f : \mathcal{H} \to \mathcal{D}$ is a CLGF then $f_{CU} : \mathcal{H} \to \mathcal{D}_{CU}$, defined as $f_{CU}(x) = (f(x), x)$, is completely unimodal. If $x \in \mathcal{H}$ is the maximum (minimum) of f_{CU} then x is a minimum of f.*

Proof. First note that maxima of f_{CU} must be maxima of f, since $f_{CU}(x) \geq f_{CU}(y) \implies f(x) \geq f(y)$ for any neighbors x, y.

By Theorem 2.2, it suffices to prove that f_{CU} restricted to any two-dimensional face has a unique local maximum. Consider any two-dimensional face F and assume f_{CU} has more than one local maximum on F. These cannot be neighbors, since neighbors have comparable and different values. Thus they are on the diagonal. By Corollary 7.2, all values of f on F are equal, so Case 2 of Definition 9.1 applies whenever the value of two neighbors x, y on F are compared. Hence the coordinate where x and y are different determines the order, and it follows that any local maximum has both F-coordinates set to 1. But there is only one such corner, contradicting our assumption that there are more than one maxima. This completes the proof. $\square$

Note that the reduction is efficient: f_{CU} can be computed in a single vertex of $\mathcal{H}$ using only one call to f.

10 Reducing CLGFs to LP-type problems

Local improvement algorithms are well-studied since the 1950's in the context of the simplex method for linear programming. The subexponential algorithms we present in this paper and in [6] are adaptations of algorithms originally suggested for linear programming [16, 19, 20]. The papers [22, 19, 11] present an abstract framework called *LP-type problems*, appropriate for several problems, including linear programming, MiniBall (given a set of points in d-space, determine the center and radius of the smallest enclosing ball), PolyDist (given two convex polyhedra in d-space, determine the distance between them) [11], and others [19]. We now show how optimizing a CLG-structure can be reduced to solving an LP-type problem. Together with the results of Section 8, this provides an interesting link between game theory and computational geometry, and allows to transfer any possible new algorithms for LP-type problem to CLG-optimization and games. We start by briefly defining the abstract framework [22]; see [22, 19] for details.

Definition 10.1 (LP-type problem) *Let H be a finite set, $\mathcal{W}$ a linearly ordered set, and $w : 2^H \to \mathcal{W}$ a function. The pair (H, w) is called an* LP-type problem *if it satisfies the following properties.*

$$F \subseteq G \subseteq H \quad \Longrightarrow \quad w(F) \leq w(G) \tag{1}$$

$$\left.\begin{array}{c} F \subseteq G \subseteq H \\ w(F) = w(G) \\ h \in H \end{array}\right\} \quad \Longrightarrow \quad \left\{\begin{array}{c} w(F \cup h) > w(F) \\ \Longleftrightarrow \\ w(G \cup h) > w(G) \end{array}\right. \tag{2}$$

A *basis* B for an LP-type problem is a subset of H with $w(B') < w(B)$ for all $B' \subset B$. A *basis for* $G \subseteq H$ is a basis $B \subseteq G$ with $w(B) = w(G)$. To 'solve' an LP-type problem means to find its basis.

The following theorem and proof describe the reduction.

Theorem 10.2 *A CLGF $f : \mathcal{P} \to \mathcal{D}$, where $\mathcal{D}$ is linearly ordered, can be reduced to an LP-type problem such that a basis for the LP-type problem defines a maximum of the CLGF.*

Proof. Recall that $\mathcal{P} = \prod_{j=1}^{d} \mathcal{P}_j$, where $\mathcal{P}_j = \{e_{j,1}, \ldots, e_{j,\delta_j}\}$ and all $e_{j,i}$ are distinct. Let H be the set of all $e_{j,i}$. For $G \subseteq H$ such that $G \cap \mathcal{P}_j \neq \emptyset$ for all j, define the associated structure of G as $\sigma(G) = \prod_{j=1}^{d}(\mathcal{P}_j \cap G)$, clearly a substructure of $\mathcal{P}$. The value set of our LP-type problem will be $\mathcal{W} := \mathcal{D} \cup \mathbb{N}$, with the usual orders on $\mathcal{D}$ and on $\mathbb{N}$, and all elements of $\mathbb{N}$ smaller than those of $\mathcal{D}$. Now define $w : 2^H \to \mathcal{W}$ as follows:

$$w(G) = \begin{cases} |G| \in \mathbb{N} & \text{if } G \cap \mathcal{P}_j = \emptyset \text{ for some } j; \\ \max\{f(x) : x \in \sigma(G)\} \in \mathcal{D} & \text{otherwise.} \end{cases}$$

I.e., if G contains at least one element from each $\mathcal{P}_j$ then $w(G)$ is the maximum of all elements in the associated substructure; otherwise $w(G) = |G|$.

Property (1) of Definition 10.1 is immediate: if $w(F) \in \mathcal{D}$ then $\sigma(F)$ is a substructure of $\sigma(G)$, so the maximum in $\sigma(F)$ is no bigger than the maximum in $\sigma(G)$. If $w(F) \in \mathbb{N}$ then $w(G)$ is either a bigger natural number or an element of $\mathcal{D}$. Proving (2) needs some more work.

Let $F \subseteq G \subseteq H$ be such that $w(F) = w(G)$ and take $h \in H$. We need to prove that $w(F \cup h) > w(F) \iff w(G \cup h) > w(G)$. If $w(F) \in \mathbb{N}$, then $F = G$, since otherwise $w(G)$ would be greater than $w(F)$, and the equivalence follows. Assume $w(F) \in \mathcal{D}$. The left-to-right implication is obvious: if $w(F) \in \mathcal{D}$ then $\sigma(F \cup h)$ is a substructure of $\sigma(G \cup h)$. Therefore $w(F \cup h) \leq w(G \cup h)$, and we get $w(G \cup h) \geq w(F \cup h) > w(F) = w(G)$.

Now suppose $w(G \cup h) > w(G)$. It remains to show that $w(F \cup h) > w(F)$. Assume the contrary and let $x \in \sigma(F)$ be a local maximum on $\sigma(F)$. Note that $w(G) = w(F) = w(F \cup h)$ by assumptions ($w(F)$ cannot be greater than or incomparable to $w(F \cup h)$ by Property (1) of Definition 10.1). Since any neighbor of x on $\sigma(G \cup h)$ is an element of either $\sigma(F \cup h)$ or $\sigma(G)$, they all have smaller or equal values, so x is a local maximum on $\sigma(G \cup h)$. This contradicts the assumption that $\sigma(G \cup h)$ has a greater local maximum and concludes the proof. $\square$

Although the theorem requires a linear order, this is not a major drawback. First, many algorithms solving LP-type problems do not rely on the order being total. Second, the order on strategy values for parity games can easily be made total. Rather than comparing values of strategies componentwise in the tuple of vertex values, comparing them lexicographically extends the order to a total one.

11 Conclusions

We defined the class of Completely Local-Global functions, capturing optimization structure underlying parity and simple stochastic games. The main motivation is that solving parity and simple stochastic games amounts to optimizing CLGFs. By abstracting away from obscuring details, analysis and development of algorithms for solving games is made easier. This also allowed us to adapt many algorithms from pseudo-Boolean optimization and linear programming to games.

CLGFs possess properties that make several attractive optimization algorithms applicable. Furthermore, any CLGF on hypercubes can be reduced to a completely unimodal function, and the results for this class can be applied.

By adapting an algorithm for linear programming [20], we give a new simple subexponential randomized algorithm for optimizing CLGFs, and thereby also for solving parity and simple stochastic games. We also described a generic reduction from CLGFs to LP-type problems, thus any possible future algorithms for such problems apply.

CLG-structures simultaneously admit subexponential and multiple switch algorithms. Although the latter currently have worse known upper bounds, good practical behavior of such algorithms, confirmed by experiments, make them very attractive and competitive [2, 5, 3]. Hopefully a thorough investigation of random walks on the favorable CU- or CLG-structures will allow for improved bounds for multiple switching algorithms, both on the average and in the worst case.

References

[1] D. Aldous. Minimization algorithms and random walk on the d-cube. *The Annals of Probability*, 11(2):403–413, 1983.

[2] H. Björklund, V. Petersson, and S. Vorobyov. Experiments with iterative improvement algorithms on completely unimodal hypercubes. Technical Report 2001-017, Information Technology/Uppsala University, September 2001. `http://www.it.uu.se/research/reports/`.

[3] H. Björklund, S. Sandberg, and S. Vorobyov. An experimental study of algorithms for completely unimodal optimization. Technical Report 2002-030, Department of Information Technology, Uppsala University, October 2002. `http://www.it.uu.se/research/reports/`.

[4] H. Björklund, S. Sandberg, and S. Vorobyov. Memoryless determinacy of parity games: A simple proof. Technical Report 2002-033, Department of Information Technology, Uppsala University, October 2002. `http://www.it.uu.se/research/reports/`, submitted for publication.

[5] H. Björklund, S. Sandberg, and S. Vorobyov. Optimization on completely unimodal hypercubes. Technical Report 018, Uppsala University / Information Technology, May 2002. `http://www.it.uu.se/research/reports/`.

[6] H. Björklund, S. Sandberg, and S. Vorobyov. A discrete subexponential algorithm for parity games. In H. Alt, editor, *20th International Symposium on Theoretical Aspects of Computer Science, STACS'2003*, Lecture Notes in Computer Science, Berlin, March 2003. Springer-Verlag. To appear. Full preliminary version: TR-2002-026, Department of Information Technology, Uppsala University, September 2002.

[7] E. Boros and P. L. Hammer. Pseudo-boolean optimization. Technical Report RRR 48-2001, RUTCOR Rutger Center for Operations Research, 2001.

[8] A. Condon. The complexity of stochastic games. *Information and Computation*, 96:203–224, 1992.

[9] A. Condon. On algorithms for simple stochastic games. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 13:51–71, 1993.

[10] E. A. Emerson and C. S. Jutla. Tree automata, μ-calculus and determinacy. In *Annual IEEE Symp. on Foundations of Computer Science*, pages 368–377, 1991.

[11] B. Gärtner. A subexponential algorithm for abstract optimization problems. *SIAM Journal on Computing*, 24:1018–1035, 1995.

[12] P. L. Hammer, B. Simeone, Th. M. Liebling, and D. De Werra. From linear separability to unimodality: a hierarchy of pseudo-boolean functions. *SIAM J. Disc. Math.*, 1(2):174–184, 1988.

[13] A. J. Hoffman and R. M. Karp. On nonterminating stochastic games. *Management Science*, 12(5):359–370, 1966.

[14] D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37:79–100, 1988.

[15] M. Jurdziński. Deciding the winner in parity games is in UP∩co-UP. *Information Processing Letters*, 68:119–124, 1998.

[16] G. Kalai. A subexponential randomized simplex algorithm. In *24th ACM STOC*, pages 475–482, 1992.

[17] W. Ludwig. A subexponential randomized algorithm for the simple stochastic game problem. *Information and Computation*, 117:151–155, 1995.

[18] Y. Mansour and S. Singh. On the complexity of policy iteration. In *Uncertainty in Artificial Intelligence'99*, 1999.

[19] J. Matoušek, M. Sharir, and M. Welzl. A subexponential bound for linear programming. In *8th ACM Symp. on Computational Geometry*, pages 1–8, 1992.

[20] J. Matoušek, M. Sharir, and M. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16:498–516, 1996.

[21] V. Petersson and S. Vorobyov. A randomized subexponential algorithm for parity games. *Nordic Journal of Computing*, 8:324–345, 2001.

[22] M. Sharir and E. Welzl. A combinatorial bound for linear programming and related problems. In *9th Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 577 of *Lecture Notes in Computer Science*, pages 569–579, Berlin, 1992. Springer-Verlag.

[23] C. A. Tovey. Low order polynomial bounds on the expected performance of local improvement algorithms. *Mathematical Programming*, 35:193–224, 1986.

[24] C. A. Tovey. Local improvement on discrete structures. In E. Aarts and Lenstra J. K., editors, *Local Search in Combinatorial Optimization*, pages 57–89. John Wiley & Sons, 1997.

[25] J. Vöge and M. Jurdziński. A discrete strategy improvement algorithm for solving parity games. In *CAV'00: Computer-Aided Verification*, volume 1855 of *Lect. Notes Comput. Sci.*, pages 202–215. Springer-Verlag, 2000.

[26] D. Wiedemann. Unimodal set-functions. *Congressus Numerantium*, 50:165–169, 1985.

[27] K. Williamson Hoke. Completely unimodal numberings of a simple polytope. *Discrete Applied Mathematics*, 20:69–81, 1988.

[28] M. Yannakakis. Computational complexity. In E. Aarts and Lenstra J. K., editors, *Local Search in Combinatorial Optimization*, pages 19–55. John Wiley & Sons, 1997.

[29] W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theor. Comput. Sci.*, 200:135–183, 1998.

Recent technical reports from the Department of Information Technology

UPPSALA UNIVERSITET

January 2003
ISSN 1404-3203
http://www.it.uu.se/