On the Smoothed Complexity of Combinatorial Local Search

Yiannis Giannakopoulos
School of Computing Science, University of Glasgow, UK

Alexander Grosz
School of Computation, Information and Technology, Technical University of Munich, Germany

Themistoklis Melissourgos
School of Computer Science and Electronic Engineering, University of Essex, UK

Abstract

We propose a unifying framework for smoothed analysis of combinatorial local optimization problems, and show how a diverse selection of problems within the complexity class PLS can be cast within this model. This abstraction allows us to identify key structural properties, and corresponding parameters, that determine the smoothed running time of local search dynamics. We formalize this via a black-box tool that provides concrete bounds on the expected maximum number of steps needed until local search reaches an exact local optimum. This bound is particularly strong, in the sense that it holds for any starting feasible solution, any choice of pivoting rule, and does not rely on the choice of specific noise distributions that are applied on the input, but it is parameterized by just a global upper bound ϕ on the probability density. The power of this tool can be demonstrated by instantiating it for various PLS-hard problems of interest to derive efficient smoothed running times (as a function of ϕ and the input size).

Most notably, we focus on the important local optimization problem of finding pure Nash equilibria in Congestion Games, that has not been studied before from a smoothed analysis perspective. Specifically, we propose novel smoothed analysis models for general and Network Congestion Games, under various representations, including explicit, step-function, and polynomial resource latencies. We study PLS-hard instances of these problems and show that their standard local search algorithms run in polynomial smoothed time.

Further applications of our framework to a wide range of additional combinatorial problems can be found in the full version of our paper.

2012 ACM Subject Classification Theory of computation → Algorithmic game theory; Mathematics of computing → Combinatorial optimization; Theory of computation → Computational complexity and cryptography

Keywords and phrases Smoothed Analysis, local search, better-response dynamics, PLS-hardness, combinatorial local optimization, congestion games, pure Nash equilibria

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.72

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2211.07547` [12]

Funding *Alexander Grosz*: Supported by the Alexander von Humboldt Foundation with funds from the German Federal Ministry of Education and Research (BMBF).

Acknowledgements Y. Giannakopoulos is grateful to Diogo Poças for many useful discussions and inspiration during the early stages of this project.

51st International Colloquium on Automata, Languages, and Programming (ICALP 2024).
Editors: Karl Bringmann, Martin Grohe, Gabriele Puppis, and Ola Svensson;
Article No. 72; pp. 72:1–72:19

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Local search heuristics are some of the most prominent, and widely used in practice, algorithms for solving computationally hard, combinatorial problems [1, 15]. Their appeal stems not only from their simplicity and theoretical elegance, but also from the fact that, for many applications, they seem to perform remarkably well both in terms of their running time and the quality of the solutions they produce.

Theoreticians have long tried to rigorously study the performance of local search, but also explain its prevalence in practice. Johnson, Papadimitriou, and Yannakakis [13] introduced the class PLS to capture the complexity of local optimization problems; since then, many important such problems have been shown to be PLS-complete, implying that they most likely cannot be solved (exactly) in polynomial time. This hardness applies not only to local search algorithms, but to arbitrary local optimization methods. For local search, in particular, PLS-hardness (under tight reductions) implies the provable existence of instances leading to exponentially slow convergence [26]. Examples include the Travelling Salesman problem under the k-Opt heuristic [14, 5] (TSP/k-Opt in the following), Local Maximum Cut on weighted graphs [23] (LocalMaxCut), and the problem of finding pure Nash equilibria in Congestion Games [10] (PNE-Congestion).

On the other hand, Orlin, Punnen, and Schulz [17] designed a local-search-based polynomial-time scheme for efficiently computing *approximately* locally-optimal solutions for general combinatorial optimization problems (with linear objectives). Although this result provides concrete justification for the practical tractability of local optimization, there are still many important aspects that call for further investigation. First, if one requires exponential accuracy, their FPTAS still cannot provide polynomial running times. Secondly, approximate solutions do not always make sense for all local optimization problems; there are problems in PLS that are inherently exact and they are not derived by simply considering the local version of some "master" global optimization problem. A notable example is PNE-Congestion [19]. Finally, we would like to be able to argue about the more general family of "vanilla" local search, and to ideally get positive results that do not depend on additional details and specific choices of pivoting rules. Addressing these points is a key objective of the present paper.

Smoothed analysis was introduced by Spielman and Teng [24] as a more realistic alternative to traditional worst-case analysis, where now the adversarially selected input is submitted to small random shocks of its numerical parameters, *before* being presented to the algorithm; the running time is then measured *in expectation* with respect to these perturbations. Under this model, Spielman and Teng were able to show that Simplex, the archetypical method for solving linear programs, is guaranteed to terminate in polynomial time (under a shadow pivoting rule) – as opposed to its exponential complexity under worst-case analysis. This remarkable result established smoothed analysis as a canonical framework for studying the performance of algorithms beyond the worst-case (see [20] for an overview of this field).

In particular, smoothed analysis has been applied successfully to important local search algorithms, providing thus a theoretical basis for the justification of their good performance in practice; these include, e.g., the k-Opt heuristic for TSP and the Flip heuristic for LocalMaxCut. (A more detailed exposition of related work on this front is deferred to the following sections of this paper, where each of our local optimization problems of interest is explicitly studied; see, namely, Section 4 and [12, Sec. 5].) A common characteristic of this prior work, though, is that running-time analysis is usually tailored specifically to the local optimization problem at hand. This naturally creates the need for technically

heavy derivations, from which it is generally not clear how to pin down the core properties of the underlying local-search structure that allow for the efficient smoothed complexity. Furthermore, as a result, it is often not easy to immediately generalize these results to capture interesting extensions, e.g., argue about the *asymmetric* version of TSP, or go from TSP/2-Opt to TSP/k-Opt. Finally, this lack of sufficient abstraction is one of the reasons that smoothed analysis has not been yet considered at all for prominent PLS-hard problems, including, e.g., PNE-Congestion. Dealing with this set of challenges is another driving force behind our paper.

Our Results and Outline

We start by proposing an abstract model for smoothed analysis of combinatorial local optimization (CLO) in Section 2. Our family of CLO problems includes problems in PLS that have an arbitrary combinatorial neighbourhood structure and linear objective functions; essentially, our model generalizes that of [17] beyond binary configurations. In Section 2.1, we add our smoothness layer that introduces probabilistic noise (independently) to the cost parameters of the CLO problem. No further assumptions are made on the distributions of the perturbed costs; their densities are only parameterized by a global upper bound of ϕ. This is a standard model (employed, e.g., in [3, 22, 8, 9]) that makes positive smoothed-analysis results even stronger, and extends the seminal model of Gaussian perturbations from [24].

Section 3 contains our key technical result for deriving upper bounds on the expected (under smoothness) number of local-search steps, until an *exact* local optimum is reached. Our black-box tool (Theorem 2) can be readily applied to an abstract CLO problem, once its underlying neighbourhood structure is appropriately captured; this is formalized through the notion of *separability* (see Definition 1), quantified by a collection of three parameters that are critical for the upper bound given by Theorem 2. We note here that our bounds are robust against the specific choice of a starting point for the local search dynamics, as well as the pivoting rule utilized at every step to transition to an improving neighbour. In other words, our main black-box tool establishes bounds for the entire *family* of local search heuristics of a local optimization problem. At a technical level, our proof works by lower-bounding the probability that *all* steps of the local-search sequence improve sufficiently the objective.

In Section 4 we demonstrate the applicability of our general framework by instantiating it for PNE-Congestion, a prominent PLS-complete problem which has not been studied before from a smoothed analysis perspective. First, we propose smoothed analysis formulations for various representations of interest for the problem, namely explicit, step-function, or polynomial resource latencies. Next, after formally establishing how PNE-Congestion is indeed a CLO problem in Section 4.1, we identify a natural parameterization of the problem that we call B-*restrained games* (Section 4.2), where B is an upper bound on the number of resources that can be changed during a single-player deviation (see Definition 5). Interestingly enough, the case of constant B is still rich enough to encode the full PLS-hardness of PNE-Congestion ([12, Appendix B]), while at the same time it can be shown (see the proof of Theorem 6) to be appropriately separable in order to immediately provide polynomial smoothed running time bounds via our black-box tool developed in Section 3. Similarly, in Section 4.3 we also study a special class of *network* congestion games, which we term (A, B)-*compact* (Definition 9); we establish polynomial smoothed complexity for various families of instances, including the one where A is polynomial and B is constant (Corollary 11), which we pair with a complementing PLS-hardness proof ([12, Appendix C]).

Finally, we apply our high-level framework to various other local optimization problems of interest, by first formally establishing that they can be viewed as CLO problems and then identifying the proper separability structures that can be plugged into our black-box tool to

provide good smoothed bounds for local search. This not only unifies, and greatly simplifies, prior existing positive results, but also allows us to extend or improve them. Notable examples include: rederiving and strengthening the polynomial smoothed time for local Max-Cut, for graphs with up to logarithmic degree, first given by Elsässer and Tscheuschner [7]; extending the polynomial smoothed time, of Englert, Röglin, and Vöcking [8], for the 2-Opt heuristic for the general (symmetric) Travelling Salesman problem (TSP), to k-Opt neighbourhoods and to *asymmetric* TSP (ATSP); and improving the quasipolynomial smoothed time for Network Coordination Games, by Boodaghians, Kulkarni, and Mehta [4], to polynomial for constant-degree graphs.

Due to space constraints, the study of all these additional applications of our framework is deferred to the full version of our paper [12].

2 Smoothed Combinatorial Local Optimization

In this section we formalize our model and fix the necessary notation.

We denote by $\mathbb{N}$ and $\mathbb{R}$, the natural and real numbers, respectively. We also denote $\mathbb{N}^* \coloneqq \mathbb{N} \setminus \{0\}$ and $[k] \coloneqq \{1, 2, \ldots, k\}$, $[\![k]\!] \coloneqq \{0\} \cup [k]$ for $k \in \mathbb{N}$. We will use boldface notation for vectors, $\boldsymbol{s} = (s_1, s_2, \ldots, s_n) \in \mathbb{R}^n$. For an index $i \in [n]$, we use $\boldsymbol{s}_{-i}$ to denote the $(n-1)$-dimensional vector that results from an n-dimensional vector $\boldsymbol{s}$ if we remove its i-component; in that way, we can express $\boldsymbol{s}$ as $(s_i, \boldsymbol{s}_{-i})$ in order to easily denote deviations in the i-th component. More generally, for a set of indices $I \subseteq [n]$, $\boldsymbol{s}_I$ denotes the $|I|$-dimensional vector that we get if we keep only the components of $\boldsymbol{s}$ whose indices are in I. For finite sets $I, \mathcal{I}$, we say that $\mathcal{I}$ is a *cover* of I if $I \subseteq \bigcup_{I' \in \mathcal{I}} I'$. All logarithms appearing in our paper are of base 2.

Combinatorial local optimization (CLO)

An instance of a *combinatorial local optimization (CLO)* problem is composed of:

- A set of feasible *configurations* $\boldsymbol{S} \subseteq [\![M]\!]^\nu \times \{0,1\}^{\bar{\nu}}$, where $M, \nu \in \mathbb{N}^*$, $\bar{\nu} \in \mathbb{N}$. A configuration $\boldsymbol{s}$ can be expressed as $\boldsymbol{s} = (\boldsymbol{s}^\bullet, \boldsymbol{s}^\circ)$, where $\boldsymbol{s}^\bullet = (s_1, \ldots, s_\nu) \in [\![M]\!]^\nu$ is called its *cost part* and $\boldsymbol{s}^\circ = (s_{\nu+1}, \ldots, s_{\nu+\bar{\nu}}) \in \{0,1\}^{\bar{\nu}}$ its *non-cost part*.
- A vector of *costs* $\boldsymbol{c} = (c_1, c_2, \ldots, c_\nu) \in [-1, 1]^\nu$. We call the c_i *cost coefficients*.[1]
- A *neighbourhood* function $N : \boldsymbol{S} \longrightarrow 2^{\boldsymbol{S}}$. For $\boldsymbol{s} \in \boldsymbol{S}$, any configuration $\boldsymbol{s}' \in N(\boldsymbol{s})$ will be called a *neighbour* of $\boldsymbol{s}$.

For the special case of $M = 1$ we will call our problem *binary*.

The *cost* of a configuration $\boldsymbol{s}$ (with respect to a fixed cost vector $\boldsymbol{c}$) is given by

$$C(\boldsymbol{s}) \coloneqq \boldsymbol{c} \cdot \boldsymbol{s}^\bullet = \sum_{i=1}^{\nu} c_i s_i.$$

[1] Note that the restriction of costs to the range of $[-1, 1]$ has been made to facilitate the translation to smoothed CLO problems below (see Section 2.1). In fact, due to the linearity of our objective, scaling $\boldsymbol{c}$ by any positive constant does not change the structure of the problem, and we will implicitly use this fact when formulating other problems as CLO problems.

A configuration $\boldsymbol{s}$ is said to be a *local optimum (minimum*[2]*)* if there are no neighbours with better costs; formally,

$$C(\boldsymbol{s}) \leq C(\boldsymbol{s}') \qquad \text{for all } \boldsymbol{s}' \in N(\boldsymbol{s}). \tag{1}$$

For every CLO problem we can define its *configuration* (or *neighbourhood*) *graph*, made up by all edges pointing from configurations to their neighbours. Formally, it is the directed graph $G = (V, E)$ with node set $V = \boldsymbol{S}$ and edges $E = \{(\boldsymbol{s}, \boldsymbol{s}') \mid \boldsymbol{s} \in \boldsymbol{S},\ \boldsymbol{s}' \in N(\boldsymbol{s})\}$. Observe that the configuration graph does not depend on the costs $\boldsymbol{c}$, but only on the combinatorial structure of the problem. Taking the cost vector $\boldsymbol{c}$ into consideration, we can now restrict the configuration graph edges to those that correspond to locally improving moves, i.e. take $E' = \{(\boldsymbol{s}, \boldsymbol{s}') \in E \mid C(\boldsymbol{s}') < C(\boldsymbol{s})\}$. The resulting acyclic subgraph $G' = (V, E')$ is called *transition graph*.

The transition graph provides an elegant and concise interpretation of local optimization: finding a local optimum of a combinatorial (local optimization) problem translates to finding a sink of its transition graph. This task is exactly the object of interest of our paper: we study (the running time of) algorithms that find local optima of such problems.

Complexity

If the configuration set and the neighbourhoods $N(\cdot)$ were given explicitly in the input of a local optimization algorithm, then finding a locally optimal solution would have been a computationally trivial task: we could afford (in polynomial time) to exhaustively go over all nodes of the transition graph, until we find a sink. Notice that the existence of a sink is guaranteed, by the fact that $\boldsymbol{S}$ is finite. However, most problems of interest (including all the problems that we study here) have exponential-size configuration graphs, with respect to the dimension $\nu + \bar{\nu}$ of the problem. Therefore, $\boldsymbol{S}$ and N are usually instead described *implicitly*, via some succinct representation (that is polynomial on ν, $\bar{\nu}$ and M). Then, the computational complexity of our algorithms is naturally measured as a function of the critical parameters ν, $\bar{\nu}$ and M (and the bit representation of the costs $\boldsymbol{c}$).

Polynomial local optimization (PLS)

In this paper we want to study problems contained in PLS, the "canonical" complexity class for local optimization problems, introduced by Johnson, Papadimitriou, and Yannakakis [13]. Therefore, without further mention, from now on we will assume that our CLO problems further satisfy the following properties:

- An *initial* configuration $\boldsymbol{s}_0 \in \boldsymbol{S}$ can be computed in polynomial time (on the size of the input).
- There exists a polynomial-time algorithm that, given as input any configuration $\boldsymbol{s}$, decides whether $\boldsymbol{s}$ is a local optimum and, if not, returns an improving neighbour $\boldsymbol{s}' \in N(\boldsymbol{s})$ with $C(\boldsymbol{s}') < C(\boldsymbol{s})$. Such an algorithm is called *pivoting rule*.

It is important to clarify that the polynomial time algorithms of the bullets above are formally *not* part of the description of the CLO problem, neither are they required to be specified in the input of a local optimization algorithm. The *existence* of these algorithms is

[2] Here we choose to write the local optimality condition with respect to minimization problems. All the theory developed in this paper applies immediately to local maximization problems as well: just flip the inequality in (1) or, equivalently, negate the cost vector $\boldsymbol{c}$.

merely a requirement for the *membership* of the problem in the class PLS (similarly to the existence of short certificates and an efficient verifier for membership in NP). As a result, PLS can be interpreted as the class of problems that correspond to looking for a sink in an (implicitly given, possibly exponentially large) directed acyclic graph[3], where at least one node can be found efficiently and, for any given node, at least one neighbour (if such exists) can be found efficiently [6, 11].

This interpretation naturally gives rise to the *standard local search* heuristic: start from an arbitrary initial configuration, and at every iteration perform an (arbitrary[4]) locally improving move until no such move exists anymore. For a fixed pivoting rule, this corresponds to traversing a single path of the transition graph. Due to the definition of PLS membership above, the running time of such a process is thus determined by the *number* of these local search iterations. In the worst case, this amounts to bounding the length of the *longest path* in the transition graph.

It is important to emphasize that, regardless of being a very natural heuristic, standard local search is definitely not the only method for finding local optima: as a matter of fact, we know that there exist local optimization problems that are efficiently solvable via more involved, "centralized" methods (e.g., by using linear programming), but for which standard local search would provably require exponential time (see, e.g., [2]): more precisely, there exist nodes in their transition graphs from which *all* paths have exponential length [23].

To assert the intractability of a problem, the common argumentation is via complexity theory techniques that prove the conditional inexistence of polynomial time algorithms. If a problem is PLS-hard, then unless PLS = P, there is no algorithm that solves it in polynomial time under traditional worst-case analysis. However, since we are studying the performance of local search heuristics, we are in fact interested in specifically *their* running time. Schäffer and Yannakakis [23] introduced the notion of a stronger reduction among problems in PLS, named *tight* PLS*-reduction*, which preserves key structural properties of the initial instance to the target-problem's instance. An important implication of an initial-problem P having a tight PLS-reduction to target-problem Q is the following: for any instance I of P, a path in the transition graph of the corresponding reduced instance J of Q induces a path of I of length no larger than its own. In particular, if for an instance I there exists a starting configuration from which all paths to solutions are exponentially long, this also applies to the reduced instance J. We show that all the problems we study have this property, either by referring to explicit bad instances for the problem at hand, or through (chains of) tight reductions, which we construct if they haven't been established before.

For a more thorough treatment of the complexity of local optimization and the class PLS, the interested reader is referred to the excellent monograph of Yannakakis [26].

2.1 Smoothed Combinatorial Local Search

Under traditional, worst-case algorithmic analysis, the running time of an algorithm for a CLO problem would be evaluated against an adversarially selected cost vector $\boldsymbol{c} = (c_1, \ldots, c_\nu)$. Instead, our goal here is to propose a systematic *smoothed analysis* [25, 21, 20] framework for local optimization. Therefore, for the remainder of this paper we assume that $\boldsymbol{c}$ is not fixed, but drawn randomly from a product distribution. More specifically, each cost coefficient c_i is drawn independently from a continuous probability distribution with density

[3] This is the transition graph mentioned earlier.

[4] Thus, standard local search is essentially a *family* of algorithms; different pivoting rules can give rise to different local search algorithms.

$f_i : [-1, 1] \longrightarrow [0, \phi]$, where $\phi \geq \frac{1}{2}$. These distributions can be adversarially selected, but their realizations c_i are not; the running time is then computed *in expectation* with respect to the random cost vector $\boldsymbol{c}$. An efficient algorithm runs in time polynomial in the combinatorial-structure parameters ν, $\bar{\nu}$ and M of the problem, and in the smoothness parameter ϕ. We will sometimes refer to this model as *smoothed CLO*, if we want to give particular emphasis to the fact that we are performing a smoothed running time analysis (as opposed to worst-case analysis).

In that sense, smoothed analysis can be seen as interpolating between two extremes: an average-case analysis setting where all c_i's are drawn i.i.d. from a uniform distribution over $[-1, 1]$, derived for $\phi = \frac{1}{2}$; and traditional worst-case analysis that can be derived in the limit, via $\phi \to \infty$, as distributions f_i approximate adversarial, single-point-mass instances.

In this paper we focus on smoothed analysis for standard local search, and so our quantity of interest will be the expected number of improving moves (for any set of adversarially given input distributions) until a local optimum is reached; that is, the expected length of the longest path in the transition graph. This allows us to also avoid some delicate representation issues that are typical of smoothed analysis, and which have to do with how the realizations c_i of *continuous* distributions (which therefore produce irrational numbers almost surely) can be handled as inputs to a Turing-machine-based computational model. For a careful discussion about this topic we point to the papers of Beier and Vöcking [3] or Röglin and Vöcking [22]. In a nutshell, for our purposes it is safe to think of the polynomial-time improving-local-move oracle (from the PLS definition) as having access to real-number arithmetic.

3 Smoothed Analysis of Local Search

In this section we present our first main result, which is a black-box tool for upper-bounding the number of improving moves of standard local search, under smoothed analysis. To achieve this, we first highlight an appropriate underlying structure of CLO problems and identify key parameters that characterize it (see Definition 1). Then, our bounds for standard local search are directly expressed as a function of these parameters (see Theorem 2). As we will demonstrate in Sections 4.2 and 4.3, and [12, Sec. 4.2], for various CLO problems of interest these parameters are well-behaved enough to result in polynomial smoothed running times.

We now introduce some terminology and notation that will be necessary for stating our main result in Theorem 2. Fix an instance of a CLO problem with cost coordinates $[\nu]$ and configurations $\boldsymbol{S}$ (see Section 2), and let $G = (\boldsymbol{S}, E)$ be its neighbourhood graph. A *covering* $(\mathcal{E}, \mathcal{I})$ of this instance consists of a cover $\mathcal{E}$ of the edges of its neighbourhood graph, and a cover $\mathcal{I}$ of its cost coordinates. That is, $\mathcal{E} \subseteq 2^E$ and $\mathcal{I} \subseteq 2^{[\nu]}$ such that $E = \bigcup_{E' \in \mathcal{E}} E'$ and $[\nu] = \bigcup_{I \in \mathcal{I}} I$. We call $\mathcal{E}$ the *transition cover* and $\mathcal{I}$ the *coordinate cover* and they contain *transition clusters* and *coordinate clusters* respectively. Recall that, under our previous discussion (see Section 2), $\mathcal{E}$ can be simply interpreted as covering all potential configuration transitions that can be made by standard local search, clustered appropriately in different groups $T \in \mathcal{E}$. For an arbitrary set of such transitions $T \subseteq E$, we also define its *core* to be the set of coordinates affected by any of the transitions:

$$\mathsf{core}(T) \coloneqq \{i \in [\nu] \mid \boldsymbol{s}_i^\bullet \neq \boldsymbol{s}_i'^\bullet \text{ for some } (\boldsymbol{s}, \boldsymbol{s}') \in T\},$$

and its *diversity* with respect to a given set of coordinates $I \subseteq [\nu]$ to be the number of different configuration changes when projected to I:

$$\delta_I(T) \coloneqq |\mathsf{range}_I(T)|, \quad \text{where } \mathsf{range}_I(T) \coloneqq \{\boldsymbol{s}_I^\bullet - \boldsymbol{s}_I'^\bullet \mid (\boldsymbol{s}, \boldsymbol{s}') \in T\}.$$

▶ **Definition 1** ((λ, β, μ)-separable instances)**.** *An instance of a CLO problem is called* (λ, β, μ)-separable *if it has a covering* $(\mathcal{E}, \mathcal{I})$ *with* $|\mathcal{E}| \leq \lambda$*, such that any transition cluster* $T \in \mathcal{E}$*:*

(a) *has a core that can be covered by using* β *many coordinate clusters from the cover* $\mathcal{I}$*; formally, there exists a* $\mathcal{I}_T \subseteq \mathcal{I}$ *with* $|\mathcal{I}_T| \leq \beta$ *such that* $\mathsf{core}(T) \subseteq \bigcup_{I \in \mathcal{I}_T} I$*, and*

(b) *has at most* μ *diversity with respect to all coordinate clusters; formally,* $\max_{I \in \mathcal{I}} \delta_I(T) \leq \mu$.

▶ **Theorem 2.** *On any* (λ, β, μ)*-separable smoothed combinatorial local optimization instance, standard local search terminates after at most*

$$3 \cdot \mu^{\beta} \lambda \cdot \nu^2 M \log(M+1) \cdot \phi$$

many steps (in expectation).

For the proof of Theorem 2 we will need the following technical lemmas. Their proofs can be found in [12, Appendix A]. We would like to highlight the fact that the structural quantities (λ, β, μ) appear as the expression $\mu^{\beta}\lambda$ in the statement of the theorem and are the deciding quantities for the running time complexity of the problem, as evidenced in the applications further below. As the choice of the covering is not unique (and often even allows for more than just one *natural* choice), this exact expression helps to explain the quantitative interaction between the properties of the covering.

▶ **Lemma 3.** *Let* $I \subseteq [\nu]$ *be a set of cost coordinates and* $\mathcal{J} \subseteq 2^{[\nu]}$ *be a cover of* I*, i.e.* $I \subseteq \bigcup_{J \in \mathcal{J}} J$*. Then, for any set of transitions* $T \subseteq E$*,*

$$\delta_I(T) \leq \prod_{J \in \mathcal{J}} \delta_J(T).$$

▶ **Lemma 4.** *Fix some* $\phi > 0$ *and let* $\boldsymbol{X} = (X_1, X_2, \ldots, X_m)$ *be a random real vector, where each component* X_i *is drawn independently from a continuous distribution with density* $f_i : \mathbb{R} \longrightarrow [0, \phi]$*. Then, for any nonzero vector* $\boldsymbol{\xi} \in \mathbb{R}^m$ *and any* $\varepsilon \geq 0$*,*

$$\mathrm{Prob}\left[0 \leq \boldsymbol{\xi} \cdot \boldsymbol{X} \leq \varepsilon\right] \leq \min\left(\frac{1}{\|\boldsymbol{\xi}\|_{\infty}}, \frac{\sqrt{2}}{\|\boldsymbol{\xi}\|_2}\right) \cdot \varepsilon\phi, \tag{2}$$

where $\|\cdot\|_2$ *and* $\|\cdot\|_{\infty}$ *denote the Euclidean and maximum norms, respectively. For the special*[5] *case where* $\boldsymbol{\xi}$ *is a (nonzero) integer vector we get*

$$\mathrm{Prob}\left[0 \leq \boldsymbol{\xi} \cdot \boldsymbol{X} \leq \varepsilon\right] \leq \varepsilon\phi. \tag{3}$$

Proof of Theorem 2. Fix a (λ, β, μ)-separable smoothed CLO instance, with neighbourhood graph $G = (\boldsymbol{S}, E)$, and let $(\mathcal{E}, \mathcal{I})$ be a covering satisfying the conditions of Definition 1.

We introduce the following notation:

$$\Delta(\boldsymbol{s}, \boldsymbol{s'}) \coloneqq \begin{cases} C(\boldsymbol{s}) - C(\boldsymbol{s'}), & \text{if this is positive,} \\ \infty, & \text{otherwise,} \end{cases}$$

[5] In this paper we will be actually making use of Lemma 4 only via its weaker bound (3), rather than the stronger form (2). This is sufficient for our purposes because, as it turns out, this has an asymptotically negligible effect on our bounds. However, we still choose to state (and prove) Lemma 4 in its full generality, since we expect it to be of potential independent interest for future extensions, especially if one considers more involved structures, or non-integral configurations.

for all $(\boldsymbol{s}, \boldsymbol{s'}) \in E$, and

$$\Delta \coloneqq \min_{(\boldsymbol{s},\boldsymbol{s'})\in E} \Delta(\boldsymbol{s}, \boldsymbol{s'}).$$

Notice that these are random variables, depending on the realizations of the cost vector $\boldsymbol{c}$.

Our first goal is to give a bound on the probability that there exists a local move that improves the cost only by (at most) $\varepsilon > 0$, as a function of this improvement bound ε. If this quantity is sufficiently small, then with high probability, standard local search will achieve improvements more than ε at *every* step, thus resulting in faster convergence. To upper bound this probability $\text{Prob}\left[\Delta \leq \varepsilon\right]$, we first use a union bound over the cover $\mathcal{E}$ of all transitions $(\boldsymbol{s}, \boldsymbol{s'}) \in E$ to get

$$\text{Prob}\left[\Delta \leq \varepsilon\right] = \text{Prob}\left[\bigcup_{(\boldsymbol{s},\boldsymbol{s'})\in E} \left[\Delta(\boldsymbol{s}, \boldsymbol{s'}) \leq \varepsilon\right]\right] \leq \sum_{T\in\mathcal{E}} \text{Prob}\left[\bigcup_{(\boldsymbol{s},\boldsymbol{s'})\in T} \left[\Delta(\boldsymbol{s}, \boldsymbol{s'}) \leq \varepsilon\right]\right]. \quad (4)$$

Next, for a fixed transition cluster $T \in \mathcal{E}$ we can express the inner union of events in (4) as

$$\begin{aligned}\bigcup_{(\boldsymbol{s},\boldsymbol{s'})\in T} \left[\Delta(\boldsymbol{s}, \boldsymbol{s'}) \leq \varepsilon\right] &= \bigcup_{(\boldsymbol{s},\boldsymbol{s'})\in T} \left[0 < \boldsymbol{c}_{\mathsf{core}(T)} \cdot \left(\boldsymbol{s}_{\mathsf{core}(T)} - \boldsymbol{s'}_{\mathsf{core}(T)}\right) \leq \varepsilon\right] \\ &= \bigcup_{\boldsymbol{x}\in\mathsf{range}_{\mathsf{core}(T)}(T)} \left[0 < \boldsymbol{c}_{\mathsf{core}(T)} \cdot \boldsymbol{x} \leq \varepsilon\right]. \quad (5)\end{aligned}$$

By the separability assumption of our CLO instance (see Definition 1), for any cluster $T \in \mathcal{E}$ there exists a subset $\mathcal{I}_T$ of $\mathcal{I}$ with $|\mathcal{I}_T| \leq \beta$ that covers $\mathsf{core}(T)$ such that, additionally, $\max_{J\in\mathcal{I}_T} \delta_J(T) \leq \max_{I\in\mathcal{I}} \delta_I(T) \leq \mu$. So, from Lemma 3 we can deduce that

$$\left|\mathsf{range}_{\mathsf{core}(T)}(T)\right| = \delta_{\mathsf{core}(T)}(T) \leq \prod_{J\in\mathcal{I}_T} \delta_J(T) \leq \mu^\beta.$$

Furthermore, observe that each $\boldsymbol{x} \in \mathsf{range}_{\mathsf{core}(T)}(T)$ is a nonzero integral vector. Thus, applying (3) of Lemma 4 we can derive that, for a fixed $\boldsymbol{x} \in \mathsf{range}_{\mathsf{core}(T)}(T)$, we have

$$\text{Prob}\left[0 < \boldsymbol{c}_{\mathsf{core}(T)} \cdot \boldsymbol{x} \leq \varepsilon\right] \leq \phi\varepsilon.$$

Therefore, using again a union bound, this time on event (5), we can see that

$$\text{Prob}\left[\bigcup_{(\boldsymbol{s},\boldsymbol{s'})\in T} \left[\Delta(\boldsymbol{s}, \boldsymbol{s'}) \leq \varepsilon\right]\right] \leq \sum_{\boldsymbol{x}\in\mathsf{range}_{\mathsf{core}(T)}(T)} \text{Prob}\left[0 < \boldsymbol{c}_{\mathsf{core}(T)} \cdot \boldsymbol{x} \leq \varepsilon\right] \leq \mu^\beta\phi\varepsilon.$$

Plugging this into (4), and using again the separability of our instance, we can finally bound our desired probability

$$\text{Prob}\left[\Delta \leq \varepsilon\right] \leq \lambda\mu^\beta\varepsilon\phi. \quad (6)$$

Now we continue with the second part of the proof, in which we utilize the probability bound in (6) to derive a concrete bound on the expected number of iterations of standard local search. Let $\mathcal{T}$ denote the random variable of that maximum length among all paths in the transition graph of our instance. Then, our goal is to bound $\mathbb{E}\left[\mathcal{T}\right]$.

Recall that the number of different possible cost-part configurations is trivially upper-bounded by $(M+1)^\nu$. Also, at every step of standard local search, the configuration cost is *strictly* decreasing. Thus $\mathcal{T} \leq (M+1)^\nu$ and, furthermore, since the range of configuration costs is within $[-M\nu, M\nu]$, and the minimum cost improvement of any iteration is Δ, we also know that $\mathcal{T} \leq \frac{2M\nu}{\Delta}$ holds. Using these, we get

$$\begin{aligned}
\mathbb{E}[\mathcal{T}] &= \sum_{t=1}^{(M+1)^\nu} \text{Prob}[\mathcal{T} \geq t] \\
&\leq \sum_{t=1}^{(M+1)^\nu} \text{Prob}\left[\Delta \leq \frac{2M\nu}{t}\right] \\
&\leq 2\lambda\mu^\beta\phi M\nu \sum_{t=1}^{(M+1)^\nu} \frac{1}{t}, && \text{due to (6)}, \\
&\leq 2\lambda\mu^\beta\phi M\nu \frac{3}{2}\log(M+1)^\nu, && \text{since } 1+\frac{1}{2}+\cdots+\frac{1}{n} \leq \frac{3}{2}\log n \ \ \forall n \geq 2, \\
&= 3\mu^\beta\lambda\phi\nu^2 M\log(M+1).
\end{aligned}$$

◀

4 Smoothed Analysis for Congestion Games

Congestion games are composed of finite nonempty sets of *players* $\mathcal{N} = [n]$ and *resources* $\mathcal{R}$. Each player $i \in \mathcal{N}$ has a *strategy set* $\Sigma_i \subseteq 2^{\mathcal{R}}$ and each resource $r \in \mathcal{R}$ has a *cost* (or *latency*) *function* $\kappa_r : [n] \longrightarrow \mathbb{R}_{\geq 0}$. Each (pure) *strategy profile* (or *outcome*) $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_n) \in \boldsymbol{\Sigma} \coloneqq \Sigma_1 \times \cdots \times \Sigma_n$ induces a *load* on each resource r, equal to the number of players that use it:

$$\ell_r(\boldsymbol{\sigma}) \coloneqq |\{i \in \mathcal{N} \mid r \in \sigma_i\}|.$$

Then, the *cost of player* i is the total cost she experiences from all resources that she is using:

$$C_i(\boldsymbol{\sigma}) \coloneqq \sum_{r \in \sigma_i} \kappa_r(\ell_r(\boldsymbol{\sigma})).$$

An (exact) pure *Nash equilibrium (PNE)* is an outcome $\boldsymbol{\sigma}^*$ from which no player can improve her cost by unilaterally deviating. Formally, for any player $i \in \mathcal{N}$ and any deviation $\sigma_i' \in \Sigma_i$:

$$C_i(\boldsymbol{\sigma}^*) \leq C_i(\sigma_i', \boldsymbol{\sigma}^*_{-i}).$$

Thus, if a strategy profile $\boldsymbol{\sigma}$ is *not* a PNE, there has to exist a player i and a deviation $\sigma_i' \in \Sigma_i$ that reduces her cost, i.e.

$$C_i(\sigma_i', \boldsymbol{\sigma}_{-i}) < C_i(\boldsymbol{\sigma}).$$

Such a strategy σ_i' is then called a *better-response* (of player i with respect to the profile $\boldsymbol{\sigma}$).

By the seminal work of Rosenthal [18] we know that function $\Phi : \boldsymbol{\Sigma} \longrightarrow \mathbb{R}_{\geq 0}$ defined by

$$\Phi(\boldsymbol{\sigma}) = \sum_{r \in \mathcal{R}} \sum_{\ell=1}^{\ell_r(\boldsymbol{\sigma})} \kappa_r(\ell) \tag{7}$$

and commonly referred to as *Rosenthal's potential*, has the property that

$$C_i(\sigma_i', \boldsymbol{\sigma}_{-i}) - C_i(\boldsymbol{\sigma}) = \Phi(\sigma_i', \boldsymbol{\sigma}_{-i}) - \Phi(\boldsymbol{\sigma}) \qquad \forall \boldsymbol{\sigma} \in \boldsymbol{\Sigma}\ \forall i \in \mathcal{N}\ \forall \sigma_i' \in \Sigma_i.$$

In other words, function Φ is an (exact) *potential* [16] of the corresponding congestion game. This implies that PNE of a congestion game correspond *exactly* to the set of *local* minimizers of its Rosenthal's potential (7), meaning that $\Phi(\boldsymbol{\sigma}^*) \leq \Phi(\sigma_i', \boldsymbol{\sigma}^*_{-i})$ for any player i and any deviation $\sigma_i' \in \Sigma_i$. This also immediately establishes the existence of PNE in *all* congestion games, since the potential function $\Phi(\boldsymbol{\sigma})$ can only take finitely many different values.

Depending on the type and representation of the cost functions, different classes of congestion games can arise. Below, we describe three prominent ones which we will focus on in this paper:

- *General.* The cost functions are given explicitly as a list of nonnegative values, one for each possible load on the resource; $(\kappa_r(1), \kappa_r(2), \ldots, \kappa_r(n))$. Notice how in this model we do not impose a monotonicity constraint; this is deliberate, to maintain full generality. If one wants to focus on nondecreasing cost functions (as is many times the case in the literature), then the step-function representation (see below) can easily be used instead.
- *Polynomials (of degree d).* The cost functions are polynomials of maximum degree $d \in \mathbb{N}$ with nonnegative coefficients. More specifically, the cost functions are given implicitly by the coefficients $\{\alpha_{r,j}\}_{r\in\mathcal{R}, j\in[\![d_r]\!]} \subseteq \mathbb{R}_{\geq 0}$, where $d_r \leq d$, via

$$\kappa_r(\ell) = \sum_{j=0}^{d_r} \alpha_{r,j}\ell^j \qquad \text{for all } \ \ell \in [n]. \tag{8}$$

- *Step functions (with d break-points).* The cost functions are nondecreasing, piecewise constant, given by pairs of break-points and value-increases. More specifically, for each resource $r \in \mathcal{R}$ there is a list of break-points $\mathcal{B}_r \subseteq [n]$ and associated jumps $\{\alpha_{r,j}\}_{j\in\mathcal{B}_r}$. We denote the number of break-points of a resource r by $d_r \coloneqq |\mathcal{B}_r|$, and we set $d \coloneqq \max_{r\in\mathcal{R}} d_r$. Then, the cost functions are given via

$$\kappa_r(\ell) = \sum_{j\in\mathcal{B}_r\cap[\ell]} \alpha_{r,j} \qquad \text{for all } \ \ell \in [n]. \tag{9}$$

To the best of our knowledge, no smoothed analysis model has been established so far for congestion games. In this paper, we propose and study the following perturbation semantics for the aforementioned classes:

- *General congestion games.* The costs $\kappa_r(\ell)$ are independently distributed according to densities $f_{r,\ell} : [0,1] \longrightarrow [0,\phi]$, for all $r \in \mathcal{R}$ and $\ell \in [n]$.
- *Polynomial games.* The coefficients $\alpha_{r,j}$ are independently distributed according to densities $f_{r,j} : [0,1] \longrightarrow [0,\phi]$, for all $r \in \mathcal{R}$ and $j \in [\![d_r]\!]$.
- *Step-function games.* The jump increases $\alpha_{r,j}$ are independently distributed according to densities $f_{r,j} : [0,1] \longrightarrow [0,\phi]$, for all $r \in \mathcal{R}$ and $j \in \mathcal{B}_r$. Notice, however, that the break-points $\mathcal{B}_r$ themselves are not subjected to any noise, and they are assumed to be fixed (and adversarially selected).

4.1 Nash Equilibria as Combinatorial Local Optimization Problems

We now show how PNE-Congestion, the problem of finding a pure Nash equilibrium in congestion games, can actually be interpreted as a combinatorial local optimization problem (with respect to our definitions in Section 2.1), for any of the cost models described above. For all models, their randomness semantics translate directly to the respective randomness of cost coefficients in the smoothed CLO problem.

- *General congestion games.* By (7) we know that PNE correspond exactly to local minimizers of the potential function $\Phi(\boldsymbol{\sigma}) = \sum_{r\in\mathcal{R}} \sum_{j=1}^{\ell_r(\boldsymbol{\sigma})} \kappa_r(j)$. Therefore, finding a PNE of a general congestion game can be viewed as a binary ($M = 1$) CLO problem, with cost dimension $\nu = |\mathcal{R}|\, n$ (the cost coordinates are given by $\mathcal{R} \times [n]$), where each strategy profile $\boldsymbol{\sigma}$ is mapped to a cost configuration $\boldsymbol{s}^\bullet = (s_{r,j})_{r\in\mathcal{R}, j\in[n]}$ given by the indicator functions:

$$s_{r,j} = \mathbb{1}\left[j \leq \ell_r(\boldsymbol{\sigma})\right]. \tag{10}$$

The CLO cost coefficients are given by $c_{r,j} = \kappa_r(j)$.

Furthermore, we want to establish a one-to-one correspondence between configurations $\boldsymbol{s}$ of the CLO problem and strategy profiles $\boldsymbol{\sigma}$ of our congestion games where, in particular, the neighbours of $\boldsymbol{s}$ are exactly the configurations corresponding to single-player deviations $\{(\sigma_i', \boldsymbol{\sigma}_{-i})\}_{i\in\mathcal{N}, \sigma_i'\in\Sigma_i}$. In that way, better-responses of the congestion game would correspond exactly to local improvements in the CLO formulation. However, this cannot be achieved by using just the cost part defined above by (10), since in that case, different strategy profiles may end up being mapped to the same cost part configuration (when they induce the same resource loads). To overcome this technical pitfall, we also maintain a non-cost part $\boldsymbol{s}^\circ$, which keeps track of the actual strategies of the players: this can be easily achieved, with only an additional polynomial size burden.[6]

Finally, notice that the neighbourhoods of the CLO problem we created can be explicitly listed and efficiently searched for a better (smaller cost) value: they have a maximum size of $n \cdot \max_{i\in\mathcal{N}} |\Sigma_i|$, which is polynomial in the description of the original congestion game.

- *Polynomial games.* Using (8), Rosenthal's potential (7) can now be written as

$$\Phi(\boldsymbol{\sigma}) = \sum_{r\in\mathcal{R}} \sum_{\ell=1}^{\ell_r(\boldsymbol{\sigma})} \sum_{j=0}^{d_r} \alpha_{r,j} \ell^j = \sum_{r\in\mathcal{R}} \sum_{j=0}^{d_r} \alpha_{r,j} \sum_{\ell=1}^{\ell_r(\boldsymbol{\sigma})} \ell^j = \sum_{r\in\mathcal{R}} \sum_{j=0}^{d_r} \alpha_{r,j} \mathfrak{S}_j(\ell_r(\boldsymbol{\sigma})),$$

where

$$\mathfrak{S}_j(\ell) \coloneqq \sum_{k=1}^{\ell} k^j \leq \ell^{j+1} \leq \ell^{d+1} \leq n^{d+1}$$

for any $\ell \in \mathbb{N}$. This induces a CLO problem with parameters

$$\nu = \sum_{r\in\mathcal{R}} (d_r + 1) \leq |\mathcal{R}|\,(d+1) \qquad \text{and} \qquad M = n^{d+1},$$

with each strategy profile $\boldsymbol{\sigma}$ corresponding to a cost configuration $\boldsymbol{s}^\bullet = (s_{r,j})_{r\in\mathcal{R}, j\in[\![d_r]\!]}$ given by:

$$s_{r,j} = \mathfrak{S}_j(\ell_r(\boldsymbol{\sigma})).$$

The costs coefficients are given by $c_{r,j} = \alpha_{r,j}$. Notice that again all neighbourhoods are efficiently searchable since they have a polynomial maximum size of $n \cdot \max_{i\in\mathcal{N}} |\Sigma_i|$.

- *Step-function games.* Using (9), Rosenthal's potential (7) can now be written as

$$\begin{aligned}\Phi(\boldsymbol{\sigma}) = \sum_{r\in\mathcal{R}} \sum_{\ell=1}^{\ell_r(\boldsymbol{\sigma})} \sum_{j\in\mathcal{B}_r\cap[\ell]} \alpha_{r,j} &= \sum_{r\in\mathcal{R}} \sum_{j\in\mathcal{B}_r\cap[\ell_r(\boldsymbol{\sigma})]} (\ell_r(\boldsymbol{\sigma}) - j + 1)\alpha_{r,j} \\ &= \sum_{r\in\mathcal{R}} \sum_{j\in\mathcal{B}_r} \max(0, \ell_r(\boldsymbol{\sigma}) - j + 1)\alpha_{r,j}.\end{aligned}$$

This induces a CLO problem with dimension

$$\nu = \sum_{r\in\mathcal{R}} |\mathcal{B}_r| = \sum_{r\in\mathcal{R}} d_r \leq |\mathcal{R}|\,d \qquad \text{and} \qquad M = n,$$

with each strategy profile $\boldsymbol{\sigma}$ corresponding to configuration $\boldsymbol{s}^\bullet = (s_{r,j})_{r\in\mathcal{R}, j\in[d_r]}$ given by

$$s_{r,j} = \max(0, \ell_r(\boldsymbol{\sigma}) - \mathcal{B}_r(j) + 1),$$

[6] E.g. we can choose $\bar{\nu} = |\mathcal{N}|$ and let $s_i^\circ \in [|\Sigma_i|]$ be the index of strategy σ_i deployed by player i in profile $\boldsymbol{\sigma}$. To simplify our exposition, in the remaining congestion game classes studied below, we will avoid explicitly discussing these non-cost parts; they are identical to the general congestion game model.

where we use $\mathcal{B}_r(j)$ to denote the j-th break-point[7] of resource r. The costs are given by $c_{r,j} = \alpha_{r,\mathcal{B}_r(j)}$. Again, it is straightforward to see that all neighbourhoods are efficiently searchable.

4.2 Restrained Congestion Games

Although congestion games are guaranteed to have (at least one) PNE, the computational problem of actually finding one is considered hard; as a matter of fact, the problem PNE-Congestion is one of the most prominent PLS-complete problems. Our goal in this section is to investigate whether this computational barrier can be bypassed, under the lens of the more optimistic complexity model of smoothed analysis.

To achieve this, we establish an upper bound (Theorem 6) on the expected number of better-responses that need to be performed until a PNE is found in a congestion game, as a function of a critical structural parameter of its action space that we identify (see Definition 5). Then, we can deduce that for congestion games in which this parameter is appropriately bounded, the smoothed running time becomes tractable (Corollary 7). At the same time, we show how such a restriction does not make the problem trivially tractable, by proving that PNE-Congestion remains PLS-complete even for this subclass of games (Theorem 8).

▶ **Definition 5** (Restrained Congestion Games). *A congestion game will be called* B-restrained*, where* $B \in \mathbb{N}$*, if the maximum number of resources changed by any single-player deviation is at most* B*. Formally,*

$$\max_{i \in \mathcal{N}} \max_{\sigma, \sigma' \in \Sigma_i} |\sigma \triangle \sigma'| \leq B,$$

where $\triangle$ *denotes the standard symmetric difference operator (recall that* σ, σ' *are subsets of resources).*

▶ **Theorem 6.** *Consider a* B*-restrained* n*-player congestion game, under any of the smoothed-analysis models described in Section 4 (namely general, polynomial, or step-function latencies), with maximum density parameter* ϕ*. Then, performing any better-response dynamics, starting from an arbitrary strategy profile, converge to an (exact) PNE of the congestion game in an expected number of iterations that is bounded by*

- $\mathcal{O}\left(n^{B+3}k^2m^2\phi\right)$ *for general games,*
- $\mathcal{O}\left(n^{B+d+2}\log(n)(d+1)^3k^2m^2\phi\right)$ *for polynomial games with degree at most* d*, and*
- $\mathcal{O}\left((d+1)^{B+2}n^2\log(n)k^2m^2\phi\right)$ *for step-function games with at most* d *break-points,*

where $m = |\mathcal{R}|$ *is the number of resources and* $k = \max_{i \in \mathcal{N}} |\Sigma_i|$ *is the maximum strategy set size.*

Proof. In Section 4.1 we already showed how congestion games can be interpreted as CLO problems. In particular, we established a one-to-one correspondence between better-responses of the players to local improvements of the CLO cost objective (which corresponds to the value of Rosenthal's potential). Using this interpretation, we can now make use of our main black-box tool from Section 3: bounding the expected number of local search steps will induce the same bound in the expected iterations of better-response dynamics in the original congestion game. Therefore, the gist of our proof is to construct coverings $(\mathcal{E}, \mathcal{I})$ so that the induced CLO problem can be shown to be (λ, β, μ)-separable (see Definition 1) for parameters with appropriately small magnitude.

[7] That is, if $\mathcal{B}_r = \{b_1, b_2, \ldots, b_{d_r}\} \subseteq [n]$, then $\mathcal{B}_r(j) = b_j$, for any $j \in [d_r]$.

We start by establishing some properties that will be shared across all three different cost models. Fix a congestion game and its corresponding CLO problem, as described in Section 4.1.[8] For convenience, we denote by $k_i \coloneqq |\Sigma_i|$ the size of the strategy set of a player i, and let $k \coloneqq \max_{i\in\mathcal{N}} k_i$. Also, in all of the models we have a similar index structure in the cost part; the cost-part is given by $(s_{r,j})_{r\in\mathcal{R},j\in J_r}$, where J_r depends on the cost function model ($J_r = [n]$ for general, $J_r = [\![d_r]\!]$ for polynomial, and $J_r = [d_r]$ for step-function costs).

The transition cover $\mathcal{E}$ is constructed from clusters that collect all edges in the neighbourhood graph (of the CLO problem) that correspond to a fixed deviation of a player, regardless of the configuration of the remaining players. Formally, we let

$$\mathcal{E} \coloneqq \{E(i,\sigma_i,\sigma_i') \mid i\in\mathcal{N},\ \sigma_i,\sigma_i'\in\Sigma_i\}, \quad \text{where}$$
$$E(i,\sigma_i,\sigma_i') \coloneqq \{(\boldsymbol{s}(\sigma_i,\boldsymbol{\sigma}_{-i}),\boldsymbol{s}(\sigma_i',\boldsymbol{\sigma}_{-i}))\in E \mid \boldsymbol{\sigma}_{-i}\in\boldsymbol{\Sigma}_{-i}\},$$

and $\boldsymbol{s}(\boldsymbol{\sigma})$ is used to denote the CLO configuration corresponding to strategy profile $\boldsymbol{\sigma}$ in the congestion game. Now we immediately get the bound

$$|\mathcal{E}| \le nk(k-1),$$

which will be used as the value for our separability parameter λ (see Definition 1).

Next, for the coordinate cover $\mathcal{I}$, we cluster the indices with respect to each resource, i.e. we choose

$$\mathcal{I} \coloneqq \{I_r \mid r\in\mathcal{R}\}, \quad \text{where} \quad I_r = \{(r,j) \mid j\in J_r\}.$$

In congestion games, a deviation $\sigma_i \to \sigma_i'$ only affects the resources $r \in \sigma_i \triangle \sigma_i'$. Their loads are changed to increase by 1 for $r \in \sigma_i' \setminus \sigma_i$ and decrease by 1 for $r \in \sigma_i \setminus \sigma_i'$; the load of all other resources does not change. Our choice of the cover $\mathcal{I}$, therefore, will allow us to settle β for all models due to the B-restrain assumption on the size of $\sigma_i \triangle \sigma_i'$. In more detail, recall that the cost parts of a configuration depend only on the loads of the resources, thus all components associated with resources $r \notin \sigma_i \triangle \sigma_i'$ (specifically, the components with coordinates I_r) remain unchanged during the transition, since the load of r does not change either. The fact that the size of those sets $\sigma_i \triangle \sigma_i'$ is universally bounded, by assumption, will let us use $\beta = B$ as a separability parameter (see Definition 1).

The remaining parameter μ depends on the structure of the configurations in the cost part. Again, we emphasize that for all cost models in Section 4.1, the sub-configuration $\boldsymbol{s}_{I_r}(\boldsymbol{\sigma})$, which comprises all components of $\boldsymbol{s}(\boldsymbol{\sigma})$ that correspond to a resource r, depends only on the load $\ell_r(\boldsymbol{\sigma})$ of resource r (under strategy profile $\boldsymbol{\sigma}$). We can therefore represent it as a function $\boldsymbol{h}_r : [\![n]\!] \to [\![M]\!]^{J_r}$, i.e., $\boldsymbol{s}_{I_r}(\boldsymbol{\sigma}) = \boldsymbol{h}_r(\ell_r(\boldsymbol{\sigma}))$. For ease of notation we write $\boldsymbol{s}_r = \boldsymbol{s}_{I_r}$ in the following.

To discuss $\mathsf{range}_{I_r}(E(i,\sigma_i,\sigma_i'))$, we need to consider the configuration changes given by

$$\boldsymbol{s}_r(\boldsymbol{\sigma}) - \boldsymbol{s}_r(\boldsymbol{\sigma}') = \begin{cases} \boldsymbol{h}_r(\ell_r(\boldsymbol{\sigma})) - \boldsymbol{h}_r(\ell_r(\sigma')) = \boldsymbol{h}_r(\ell_r(\boldsymbol{\sigma})) - \boldsymbol{h}_r(\ell_r(\boldsymbol{\sigma})+1), & r\in\sigma_i'\setminus\sigma_i, \\ \boldsymbol{h}_r(\ell_r(\boldsymbol{\sigma})) - \boldsymbol{h}_r(\ell_r(\sigma')) = \boldsymbol{h}_r(\ell_r(\boldsymbol{\sigma})) - \boldsymbol{h}_r(\ell_r(\boldsymbol{\sigma})-1), & r\in\sigma_i\setminus\sigma_i'. \end{cases}$$

The only variable in this expression is therefore the initial load $\ell_r(\boldsymbol{\sigma})$. In either case, there are n possible initial loads for each resource[9] and therefore also at most n difference-vectors within $\mathsf{range}_{I_r}(E(i,\sigma_i,\sigma_i'))$; thus $\mu \le n$. In the following, we will discuss the actual configuration difference structure for each model and whether we can improve μ over this basic bound.

[8] We will use standard notation for the various components and parameters of the game and the CLO problem, as introduced above in Section 4.

[9] Note that although there are $n+1$ different loads $0,\dots,n$, a load-increasing resource cannot already have load n and a decreasing one cannot have 0.

- *General.* The CLO representation follows a binary model with $M = 1$ and $\nu = mn$: each resource r corresponds to components $s_{r,j}$, $j = 1, \ldots, n$ (the indices from I_r), with a value of $s_{r,j} = 1$ for $j \leq \ell_r(\boldsymbol{\sigma})$ and $s_{r,j} = 0$ otherwise. The function $\boldsymbol{h}_r : [\![n]\!] \longrightarrow \{0,1\}^n$ is thus given by

  $$(\boldsymbol{h}_r(\ell))_j = \mathbb{1}[j \leq \ell], \qquad \text{for all } \; j \in [n].$$

  In particular, the vectors $\boldsymbol{\delta} \in \mathsf{range}_{I_r}(E(i, \sigma_i, \sigma_i'))$ are a result of moving the rightmost entry with value 1 within the vector $\boldsymbol{s}_r$ to the next larger or smaller load in the configuration, i.e. they are given by: $\delta_{\ell_r(\boldsymbol{\sigma})+1} = -1$ for $r \in \sigma_i' \setminus \sigma_i$; $\delta_{\ell_r(\boldsymbol{\sigma})} = +1$ for $r \in \sigma_i \setminus \sigma_i'$; and zeroes elsewhere. Since there are n many such vectors for every resource r, we cannot improve over $\mu = n$.
  In this case, thus, Theorem 2 yields an expected running time of at most

  $$3 \cdot n^B nk(k-1) \cdot (mn)^2 \cdot \phi = \mathcal{O}\left(n^{B+3} k^2 m^2 \phi\right).$$

- *Polynomial games.* For this model, each (cost-part) configuration component $s_{r,j}$ is given by accumulated monomials $\mathfrak{S}_j(\ell_r(\boldsymbol{\sigma}))$, for degrees $j = 0, 1, \ldots, d_r$. Thus, $\nu = m(d+1)$, and also $M = n^{d+1}$, in order to capture all possible values of these functions. Therefore we now get $\boldsymbol{h}_r : [\![n]\!] \longrightarrow [\![n^{d+1}]\!]^{d_r+1}$ with

  $$\boldsymbol{h}_r(\ell) = (\mathfrak{S}_0(\ell), \ldots, \mathfrak{S}_{d_r}(\ell))$$

  for the configuration component of I_r. Again, we cannot do better than the basic bound, so we use $\mu = n$.
  Similarly to the previous case for general latency functions, using Theorem 2 we can bound the expected number of better-response iterations by

  $$3 \cdot n^B nk(k-1) \cdot (m(d+1))^2 n^{d+1} \log(n^{d+1}) \cdot \phi = \mathcal{O}\left(n^{B+d+2} \log(n)(d+1)^3 k^2 m^2 \phi\right).$$

- *Step-function games.* The configuration mapping is now given by

  $$s_{r,j} = \max\left(0, \ell_r(\boldsymbol{\sigma}) - \mathcal{B}_r(j) + 1\right)$$

  and $M = n, \nu = m \cdot d$. Therefore, the (cost-part) configuration components of a resource r are represented by the function $\boldsymbol{h}_r : [\![n]\!] \longrightarrow [\![n]\!]^{d_r}$ given by

  $$\boldsymbol{h}_r(\ell) = \left(\max(0, \ell - \mathcal{B}_r(1) + 1), \ldots, \max(0, \ell - \mathcal{B}_r(d_r) + 1)\right).$$

  In this case, we can do even better than the basic $\mu = n$ bound. We investigate the structure of the differences with respect to each coordinate $j \in J_r$, for an increase in the load of a resource r, i.e. for $r \in \sigma_i' \setminus \sigma_i$ (the decreasing case follows analogously):

  $$\begin{aligned}(\boldsymbol{h}_r(\ell) - \boldsymbol{h}_r(\ell+1))_j &= \max\left(0, \ell - \mathcal{B}_r(j) + 1\right) - \max\left(0, (\ell+1) - \mathcal{B}_r(j) + 1\right) \\ &= \begin{cases} 0 - 0 = 0, & \text{if } \; \ell < \mathcal{B}_r(j) - 1, \\ -\left((\ell+1) - \mathcal{B}_r(j) + 1 - 0\right) = -1, & \text{if } \; \ell = \mathcal{B}_r(j) - 1, \\ (\ell - \mathcal{B}_r(j) + 1) - ((\ell+1) - \mathcal{B}_r(j) + 1) = -1, & \text{if } \; \ell > \mathcal{B}_r(j) - 1. \end{cases}\end{aligned}$$

  Because the jump points $\mathcal{B}_r(j)$ are ordered increasingly with respect to j, the resulting vectors $\boldsymbol{h}_r(\ell) - \boldsymbol{h}_r(\ell+1)$ are of the form $(-1, \ldots, -1, 0, \ldots, 0)$, including the zero vector. Therefore, with respect to the coordinate cluster I_r, there are at most $d_r + 1$ possible distinct vectors for the (cost-part) configuration differences, and we choose $\mu = \max_r d_r + 1 = d + 1$. By Theorem 2 we can thus bound the expected number of iterations by

  $$3 \cdot (d+1)^B nk(k-1) \cdot (m \cdot d)^2 n \log(n) \cdot \phi = \mathcal{O}\left((d+1)^{B+2} n^2 \log(n) k^2 m^2 \phi\right). \qquad \blacktriangleleft$$

An immediate corollary of Theorem 6 is that, when congestion games are sufficiently restrained, PNE can be found efficiently via better-response dynamics. In what follows, by a *constantly-* and *polylogarithmically*-restrained congestion game we mean a B-restrained game with $B \in \mathcal{O}(1)$ and $B \in \mathcal{O}(\log^c N)$ for some constant $c > 0$, respectively, where N is the size of the input.

▶ **Corollary 7.** *Better-response dynamics terminate in polynomial smoothed time for the class of constantly-restrained congestion games, and in quasipolynomial smoothed time for polylogarithmically-restrained games, under any cost model. Under the step-function cost model, in $\mathcal{O}(\log N)$-restrained congestion games with a constant number of steps d, better-response dynamics terminate in polynomial smoothed time.*

We now show that the class of constantly-restrained congestion games, for which Corollary 7 provides efficient smooth running time, constitutes a computationally meaningful restriction of arbitrary congestion games, since they can still encode the PLS-completeness of the original problem. The hardness is a straightforward adaptation of that in [19], and so the proof of Theorem 8 is deferred to [12, Appendix B] for completeness. It makes use of the fact that LocalMaxCut is PLS-complete even for constant-degree graphs.

▶ **Theorem 8.** *The problem of computing a PNE of a constantly-restrained congestion game is* PLS*-complete, for all the input models described in Section 4 (namely general, polynomial, or step-function cost representations).*

In addition to the conditional intractability that PLS-hardness implies for these families of congestion games, we show an unconditional lower bound on the worst-case running time of the standard local search algorithm of the problem. We do this by using the notion of a *tight* PLS*-reduction*, as discussed in Section 2. Since we reduce from the problem LocalMaxCut-d for $d \geq 5$ (defined in [12, Appendix B]), which admits a configuration starting from which the standard local search algorithm needs exponentially many iterations (see discussion after proof of [12, Theorem 5.10]), our tight PLS-reduction implies that standard local search of our families of congestion games – under *any* pivoting rule – takes exponential time in the worst case.

4.3 Network Congestion Games

An interesting, and very well studied, variation on the vanilla representation model for congestion games (which we presented at the start of Section 4 above) is that of *network* congestion games. In such games, the strategy sets Σ_i of the players are not given explicitly in the input, but implicitly via an underlying directed graph G whose edges constitute the resources of the game. More precisely, for each player i we are given an origin o_i and a destination d_i node of G. Then, Σ_i is defined (implicitly) as the set of all (simple) $o_i \to d_i$ paths in G. Importantly, this means that now players may have exponentially many strategies available to them.

A critical implication is that Σ_i cannot be searched *exhaustively* for better-responses. However, a better-response can still be found efficiently: keeping all other players fixed, a minimum-cost strategy of player i is a shortest path on graph G with edge costs equal to the cost $c_r(\ell_r(\boldsymbol{\sigma}_{-i}) + 1)$ of an edge/resource r when used by player i. This means that actually a polynomial-time *best*-response oracle does exist. This immediately places network congestion games in the complexity class PLS (since neighbourhoods can be searched efficiently for a local cost improvement; see our discussion in Section 2) and thus it constitutes a valid CLO

problem (via a similar interpretation as we did for general congestion games in Section 4.1). To emphasize this, we will refer to these "canonical" best-response dynamics of network congestion games as *shortest-path* dynamics.

Finding PNE of network congestion games remains a PLS-complete problem [2]. Given the prominence of these games, both in the theoretical and applied literature, in this section we want to identify conditions under which network congestion games inherit the desirable properties of their general counterparts that allow them to be tractable under smoothed analysis. More precisely, can our positive result from Theorem 6 be applied to network congestion games in a straightforward way? And which structural parameters are now relevant for the running time bound? At the same time, the network congestion game instances that allow for efficient smoothed solutions should still be interesting enough to remain PLS-hard under traditional worst-case analysis. We introduce the following family of network congestion games that are defined by two parameters.

▶ **Definition 9** (Compact Network Congestion Games). *For A, B positive integers, a network congestion game is called* (A, B)-compact *if (a) each player has at most A different best-response strategies, and (b) all such strategies are paths of length at most B. Formally, there exist strategy sets $\Sigma_i^* \subseteq \Sigma_i$, such that:*
(a) $|\Sigma_i^*| \leq A$ *for all* $i \in \mathcal{N}$ *and* $\operatorname{argmin}_{\sigma_i \in \Sigma_i} C_i(\sigma_i, \boldsymbol{\sigma}_{-i}) \subseteq \Sigma_i^*$ *for all* $i \in \mathcal{N}, \boldsymbol{\sigma}_{-i} \in \boldsymbol{\Sigma}_{-i}$, *and*
(b) $|\sigma_i| \leq B$ *for all* $i \in \mathcal{N}, \sigma_i \in \Sigma_i^*$.

Property (b) above will serve the purpose of imposing the restraint condition (see Definition 5) needed to deploy Theorem 6. Property (a) will help us constrain the exponentiality of the strategy space of network games, in order to be able to handle them using tools designed for general games. Intuitively, this property can be related to classical vehicle routing settings in which players can a priori exclude unreasonable detours or paths that involve a road with a construction site with large delay. Both properties are illuminated within the proof of our following positive result for network congestion games.

▶ **Theorem 10.** *Consider an (A, B)-compact n-player network congestion game, under any of the smoothed-analysis models described in Section 4 (namely general, polynomial, or step-function latencies), with maximum density parameter ϕ. Then, performing shortest-path dynamics, starting from an arbitrary strategy profile, converges to a PNE of the game in an expected number of iterations that is polynomial in ϕ, $(d+1)^B$, A, and the description of the game, where the parameter d depends on the cost function representation. In particular: for general latencies, $(d+1)$ can be replaced by n; for polynomial latencies, d is the maximum degree; and for step-functions, d is the maximum number of break-points.*

An immediate consequence of Theorem 10 is that (analogously to Corollary 7 for general congestion games) in (A, B)-compact network games with sufficiently small parameters A, B, a PNE can be found efficiently under smoothness.

▶ **Corollary 11.** *Let N be the size of the input, and A be a polynomial in N. Shortest-path dynamics on (A, B)-compact network congestion games under any cost model terminate in polynomial smoothed time when $B \in \mathcal{O}(1)$ and in quasipolynomial smoothed time when $B \in \mathcal{O}(\log^c N)$ for some constant $c > 0$. Under the step-function cost model, when A is a polynomial in N, $B \in \mathcal{O}(\log N)$, and the number of steps d is constant, shortest-path dynamics terminate in polynomial smoothed time.*

The following hardness result establishes that such games, even for $A, B \in \mathcal{O}(1)$, are PLS-hard. The proof is deferred to [12, Appendix C]. It is based on the reduction constructed by Ackermann, Röglin, and Vöcking [2], with special care taken in order to incorporate constant-length paths that can be established by making use of the fact that LocalMaxCut is PLS-complete even for constant degree graphs.

▶ **Theorem 12.** *The problem of computing a PNE of an (A, B)-compact network congestion game is* PLS*-complete, even for $A, B \in \mathcal{O}(1)$, for all the input models described in Section 4 (namely general, polynomial, or step-function latencies).*

Similarly to our PLS-hardness reduction of Theorem 8, the above theorem's PLS-reduction is tight, in the sense of [23] (see Section 2). The chain of tight PLS-reductions that leads to Network Congestion Games starts from LocalMaxCut with maximum degree 5, and includes Congestion Games. As we discuss after the proof of [12, Theorem 5.10], in such LocalMaxCut instances there is a starting configuration from which all improvement sequences of standard local search have exponential length in the worst case. Therefore, shortest-path dynamics on our family of network congestion games – under *any* pivoting rule – need exponentially many iterations in the worst case. In contrast, our Theorem 10 and Corollary 11 show that under smoothness, even for significantly wider instance families that include these problematic cases, shortest-path dynamics terminate after polynomially many steps in expectation.

References

1 Emile Aarts and Jan Karel Lenstra, editors. *Local Search in Combinatorial Optimization*. John Wiley, 1997.

2 Heiner Ackermann, Heiko Röglin, and Berthold Vöcking. On the impact of combinatorial structure on congestion games. *Journal of the ACM*, 55(6):1–22, 2008. doi:10.1145/1455248.1455249.

3 Rene Beier and Berthold Vöcking. Typical properties of winners and losers in discrete optimization. *SIAM Journal on Computing*, 35(4):855–881, 2006. doi:10.1137/s0097539705447268.

4 Shant Boodaghians, Rucha Kulkarni, and Ruta Mehta. Smoothed efficient algorithms and reductions for network coordination games. In *11th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 151 of *LIPIcs*, pages 73:1–73:15, 2020. doi:10.4230/LIPIcs.ITCS.2020.73.

5 Barun Chandra, Howard Karloff, and Craig Tovey. New results on the old k-opt algorithm for the traveling salesman problem. *SIAM Journal on Computing*, 28(6):1998–2029, 1999. doi:10.1137/s0097539793251244.

6 Constantinos Daskalakis and Christos Papadimitriou. Continuous local search. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2011. doi:10.1137/1.9781611973082.62.

7 Robert Elsässer and Tobias Tscheuschner. Settling the complexity of local max-cut (almost) completely. In Luca Aceto, Monika Henzinger, and Jiří Sgall, editors, *International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 171–182, 2011. doi:10.1007/978-3-642-22006-7_15.

8 Matthias Englert, Heiko Röglin, and Berthold Vöcking. Smoothed analysis of the 2-opt algorithm for the general TSP. *ACM Transactions on Algorithms*, 13(1):1–15, 2016. doi:10.1145/2972953.

9 Michael Etscheid and Heiko Röglin. Smoothed analysis of local search for the maximum-cut problem. *ACM Transactions on Algorithms*, 13(2):25:1–25:12, 2017. doi:10.1145/3011870.

10 Alex Fabrikant, Christos Papadimitriou, and Kunal Talwar. The complexity of pure nash equilibria. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 604–612, 2004. doi:10.1145/1007352.1007445.

11 John Fearnley, Spencer Gordon, Ruta Mehta, and Rahul Savani. Unique end of potential line. *Journal of Computer and System Sciences*, 114:1–35, 2020. doi:10.1016/j.jcss.2020.05.007.

12 Yiannis Giannakopoulos, Alexander Grosz, and Themistoklis Melissourgos. On the smoothed complexity of combinatorial local search, 2022. arXiv:2211.07547v3.

13 David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988. doi:10.1016/0022-0000(88)90046-3.

14 Mark W. Krentel. Structure in locally optimal solutions. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 216–221, 1989. doi:10.1109/SFCS.1989.63481.

15 Wil Michiels, Jan Korst, and Emile Aarts. *Theoretical Aspects of Local Search*. Springer, 2007. doi:10.1007/978-3-540-35854-1.

16 Dov Monderer and Lloyd S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, 1996. doi:10.1006/game.1996.0044.

17 James B. Orlin, Abraham P. Punnen, and Andreas S. Schulz. Approximate local search in combinatorial optimization. *SIAM Journal on Computing*, 33(5):1201–1214, 2004. doi:10.1137/s0097539703431007.

18 Robert W. Rosenthal. A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2(1):65–67, 1973. doi:10.1007/BF01737559.

19 Tim Roughgarden. *Pure Nash Equilibria and PLS-Completeness*, pages 261–278. Cambridge University Press, 2016. doi:10.1017/CBO9781316779309.020.

20 Tim Roughgarden, editor. *Beyond the Worst-Case Analysis of Algorithms*. Cambridge University Press, 2021. doi:10.1017/9781108637435.

21 Heiko Röglin. *The Complexity of Nash Equilibria, Local Optima, and Pareto-Optimal Solutions*. PhD thesis, RWTH Aachen University, 2008. URL: https://publications.rwth-aachen.de/record/50046.

22 Heiko Röglin and Berthold Vöcking. Smoothed analysis of integer programming. *Mathematical Programming*, 110(1):21–56, 2007. doi:10.1007/s10107-006-0055-7.

23 Alejandro A. Schäffer and Mihalis Yannakakis. Simple local search problems that are hard to solve. *SIAM J. Comput.*, 20(1):56–87, 1991. doi:10.1137/0220004.

24 Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms. *Journal of the ACM*, 51(3):385–463, 2004. doi:10.1145/990308.990310.

25 Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis: An attempt to explain the behavior of algorithms in practice. *Commun. ACM*, 52(10):76–84, 2009. doi:10.1145/1562764.1562785.

26 Mihalis Yannakakis. Computational complexity. In Emile Aarts and Jan Karel Lenstra, editors, *Local Search in Combinatorial Optimization*, chapter 2. John Wiley, 1997.