Cover-Encodings of Fitness Landscapes

Konstantin Klemm
Anita Mehta
Peter F. Stadler

SFI WORKING PAPER: 2017-04-014

Cover-Encodings of Fitness Landscapes

Konstantin Klemm klemm@ifisc.uib-csic.es
IFISC (CSIC-UIB), Campus Universitat de les Illes Balears, E-07122 Palma de Mallorca, Spain

Anita Mehta anita.mehta@cea.fr
Dipartimento di Fisica, INFM ed INFN, Universitá di Roma La Sapienza, P. A. Moro 2, 00185 Roma, Italy & Bioinformatics Group, Department of Computer Science and Interdisciplinary Center for Bioinformatics, University Leipzig, D-04107 Leipzig, Germany.

Peter F. Stadler studla@bioinf.uni-leipzig.de
Bioinformatics Group, Department of Computer Science and Interdisciplinary Center for Bioinformatics, University Leipzig, D-04107 Leipzig, Germany & Max Planck Institute for Mathematics in the Science, D-04103 Leipzig, Germany & Santa Fe Institute, Santa Fe, NM 87501, USA

Abstract

An instance of a discrete optimization problem can be tackled by a local search on a suitably defined cost or fitness landscape. Ruggedness of the landscape arrests the search or slows down its progress due to trapping in local minima. Alternatively, a heuristic approximation may be used for estimating a global cost minimum. Here we present a combination of these two approaches by *cover-encoding maps* from a larger search space to subsets of the original search space. The key idea is to construct the cover-encoding maps with the help of a suitable heuristic that singles out a near-optimal solution and results in a landscape on the larger search space that no longer exhibits trapping local minima. We present cover-encoding maps for the problems of the traveling salesman, number partitioning, maximum matching and maximum clique and demonstrate practical feasibility by simulations of adaptive walks on the encoded landscapes, finding the global minima. In addition, we discuss an analogy between certain types of encodings and coarse-graining and renormalization group of statistical physics.

Keywords

Adaptive walk, coarse graining, oracle function, genotype-phenotype map, combinatorial optimization.

1 Introduction

The performance of local search algorithms is well known to depend on the structure of the search space. The latter is determined by two largely independent ingredients: (1) A so-called encoding defines concrete representations of the configurations that are to be optimized. (2) A move set defines locality in the search space. For many of the well-studied combinatorial optimization problems and related models from statistical physics (such as spin glasses), there is a natural encoding. For instance, tours of a Travelling Salesperson Problem (TSP) are naturally encoded as permutations of the cities, and spin configurations are encoded as strings over the alphabet $\{+,-\}$ with each letter refering to a fixed spin variable. This natural encoding is usually free of redundancy or at least carries only redundancies defined by simple symmetries of the problem that could easily be factored out. For instance, TSP tours can start at any city, hence they are invariant under rotations, and many spin glass models are invariant under simultaneous flipping of all spins. This natural or "direct" encoding is often referred to as the *phenotype space*, see e.g. (Rothlauf, 2006; Neumann and Witt, 2010; Rothlauf, 2011; Borenstein and Moraglio, 2014).

For a given encoding, irrespective of whether it is genotypic or phenotypic, the performance of search crucially depends on the move set. Here, we will consider only reversible, mutation-like moves. The search space therefore is modeled as an undirected graph. More general settings are discussed e.g. by Flamm et al. (2007). The cost function assigned to a specific search space defines a *fitness landscape*. Evolutionary algorithms can thus be viewed as dynamical systems operating on landscapes. Therefore, the structure of landscapes has been studied extensively in the field (Reidys and Stadler, 2002; Østman et al., 2010; Engelbrecht and Richter, 2014).

In evolutionary computation, sometimes an additional encoding Y, the so-called *genotype space* is used (Rothlauf and Goldberg, 2003; Rothlauf, 2006). The genotype-phenotype relation is determined by a map $\alpha : Y \to X \cup \{\varnothing\}$, where $\varnothing$ represents phenotypic configurations that do not occur in the original problem, i.e., $y \in Y$ does not encode a feasible solution of the original problem whenever $\alpha(y) = \varnothing$. For example, a frequently used genotypic encoding represents TSP tours as binary strings that represent, for each of the possible adjacencies between two cities, its presence (`1`) and absence (`0`) in the tour (Applegate et al., 2006). Most binary strings to not correspond to TSP tours.

Indirect genotypic representations are usually chosen with a high degree of redundancy. Often this also introduces neutrality, i.e., adjacent configurations with the same value of the cost function. Detailed investigations of fitness landscapes from molecular biology showed that degrees of neutrality *can* facilitate optimization (Schuster et al., 1994; Reidys and Stadler, 2002) due to extensive neutral paths that prevents trapping in metastable states (Schuster et al., 1994; Fernández and Solé, 2007; Yu and Miller, 2002; Banzhaf and Leier, 2006). On the other hand, "synonymous encodings" where genotypes mapping to the same phenotype from tight clusters in the genotype space have been advocated for the design of evolutionary algorithms (Rothlauf, 2006; Choi and Moon, 2008; Rothlauf, 2011). Empirically, the introduction of arbitrary redundancy (by means of random Boolean network mapping) does not increase the performance of mutation-based search (Knowles and Watson, 2002). Klemm et al. (2012) emphasized the utility of inhomogeneous genotype-phenotype maps. If the size of the preimage $|\alpha^{-1}(x)|$ of the phenotypes is anti-correlated with the cost $f(x)$, then low cost solutions are enriched w.r.t. to genotype space and can make optimization more efficient. To enforce this anti-correlation, however, α needs to become explicitly dependent on the cost function.

In this contribution, we outline a theory of encodings that provides a recipe for constructing in a rational manner both genotype spaces and genotype-phenotype maps with desirable properties. The basic idea is not to consider the genotype-phenotype map directly. Instead the first step is the construction of a genotype space Y and an encoding scheme ϕ that maps genotypes to restrictions of the original problem rather than a particular phenotype y. The cost function then enters by guiding, for every genotype $y \in Y$ a heuristic that solves the restricted problem $\phi(y)$. This heuristic estimate then becomes the phenotype.

2 A Theory of Encoding Representations

2.1 Landscapes

An instance (X, f) of a combinatorial optimization problem consists of a finite set X and a a cost function $f : X \to \mathbb{R}$ on X. The task of the combinatorial optimization problem (X, f) is to find a *global minimum* $\hat{x} \in X$ so that $f(\hat{x}) \le f(x)$ for all $x \in X$.

A *landscape* $(X, \sim, f)$ consists of a finite set X endowed with a symmetric and irreflexive (adjacency) relation $\sim$ and cost function $f : X \to \mathbb{R}$. A point $x^* \in X$ is a strict local minimum in $(X, \sim, f)$ if (i) $f(x^*) > f(\hat{x})$ and (ii) there is no $x' \in X$ with $f(x') < f(x^*)$ and an f-non-increasing path $x^* = x_0, x_2, \dots, x_k = x'$, that is, $x_{i-1} \sim x_i$ and $f(x_{i-1}) \ge f(x_i)$ holds for $0 < i \le k$. Note that a global minimum $\hat{x}$ is not a strict local minimum as defined above.

For any $X' \subseteq X$ the restricted problem $(X', f_{|X'})$, where $f_{|X'}(x) = f(x)$ for all $x \in X'$, consists in finding a $\hat{x}' \in X'$ so that $f(\hat{x}') \le f(x')$ for all $x \in X$. A restricted landscape $(X', \sim, f_{|X'})$ can be defined analogously.

2.2 Oracle Function and Cover-Encoding Map

Definition 1. The *oracle function* $F : 2^X \to \mathbb{R}$ of an optimization problem (X, f) is

$$F(X') := \min_{x \in X'} f(x) \tag{1}$$

for all $X' \subseteq X$. We use the convention $F(\emptyset) = \infty$.

We say that a subset $X' \subseteq X$ is *good* if $F(X') = F(X)$, i.e., if X' contains a global optimum, and *bad* if $F(X') > F(X)$. The oracle function is by definition monotonic in the following sense:

$$X'' \subseteq X' \Longrightarrow F(X'') \ge F(X') \tag{2}$$

Definition 2. A function $\phi : Y \to 2^X$ is a *cover-encoding map* for X if it satisfies

(Y1) $\bigcup_{y \in Y} \phi(y) = X$.

Property (Y1) states that the collection of sets $\{\phi(y) | y \in Y\}$ is a set cover of X. The points $y \in Y$ can be thought as coding for a particular element of this set cover. In the following, we will be interested in cover-encoding maps that satisfy some or all of the following additional properties:

(Y0) $\phi(y) \neq \emptyset$.

(Y2) For every $x \in X$ there is a $y \in Y$ such that $\phi(y) = \{x\}$.

(Y3) There is $y \in Y$ such that $\phi(y) = X$.

Note that both (Y2) and (Y3) imply (Y1). Axiom (Y0) excludes infeasible points in Y.

Now consider an optimization problem (X, f) and let $\phi : Y \to 2^X$ be a cover-encoding map for X. We define $\tilde{F} : Y \to \mathbb{R}$ as the composition of ϕ with the oracle function of (X, f), i.e., $\tilde{F}(y) = F(\phi(y))$. In the following we will be interested in the relationship between the "encoded" optimization problem $(Y, \tilde{F})$ and the original problem (X, f).

If condition (Y2) is satisfied, there is $\hat{y} \in Y$ so that $\varphi(\hat{y}) = \{\hat{x}\}$ for every global optimum of the original problem. For most applications it is sufficient to find one global optimum, hence we will consider the weaker condition

(F0) There is $\hat{y} \in Y$ so that (i) $|\varphi(\hat{y})| = 1$ and $F(\varphi(y)) = f(\hat{x})$.

Condition (F0) simply states that there exists a code $y \in Y$ that identifies a global optimum of the original problem (X, f). This is sufficient to consider (X, f) and $(Y, \tilde{F})$ as "equivalent optimization problems".

In this contribution we are interested in search-based algorithms. Hence we fix an adjacency relation $\sim$ on Y. For the landscape $(Y, \sim, \tilde{F})$ we consider the following three properties:

(R1) For every $y \in Y$ with $\tilde{F}(y) = F(\hat{y})$ there is a sequence $y = y_0, y_1, \ldots, y_k = \hat{y}$ such that $y_i \sim y_{i-1}$ for $0 < i \leq k$ and $F(y_i) = F(\hat{y})$.

(R2) For every $y \in Y$ with $\tilde{F}(y) > F(\hat{y})$ there is a sequence $y = y_0, y_1, \ldots, y_k = \hat{y}$ such that $y_i \sim y_{i-1}$ for $0 < i \leq k$, $\tilde{F}(y_k) = F(\hat{y})$ and $\tilde{F}(y_{i-1}) \geq \tilde{F}(y_i)$.

(R3) Every y with $\phi(y) \neq X$ has a neighbor $y' \sim y$ with $\phi(y) \subset \phi(y')$.

Note that condition (R3) is independent of the oracle function F.

Lemma 1. (R3) implies (R2) for any oracle function F.

Proof. If $\phi(y) = X$ then $\tilde{F}(y) = F(X) = f(\hat{x}) = \tilde{F}(\tilde{y})$ by construction. Now consider an arbitrary starting point y. By (R3) there is a neighbor $y' \sim y$ such that $\phi(y) \subset \phi(y')$ and by equ.(2) we therefore have $\tilde{F}(y') \leq \tilde{F}(y)$. Repeating the argument we obtain a $\tilde{F}$-non-increasing sequence $y, y', y'', \ldots, y^{(k)}, \ldots$ along which ϕ is strictly increasing in each step. Since X is finite, there is a finite k so that $\phi(y^{(k)}) = X$ and thus $\tilde{F}(y^{(k)}) = \tilde{F}(\hat{y})$, i.e., (R2) is satisfied. □

The importance of conditions (R1) and (R2) stems from the following observation:

Theorem 1. *Suppose (X, f), $\phi : Y \to 2^X$, and the relation $\sim$ on Y are chosen such that (Y1), (F0), (R1), and (R2) are satisfied. Then the landscape $(Y, \sim, \tilde{F})$ has no strict local optimum.*

Proof. Let $y \in Y$ be an arbitrary starting point. If $\tilde{F}(y) = \tilde{F}(\hat{y})$ then y, by (R1), is not a local optimum but part of a connected neutral network that contains the global optimum $\hat{y}$. If $\tilde{F}(y) \neq \tilde{F}(\hat{y})$ then $\tilde{F}(y) > \tilde{F}(\hat{y})$. By (R2) there is a path with non-increasing values of $\tilde{F}$ that connects y to a point y' with $\tilde{F}(y') = \tilde{F}(\hat{y})$. We already know that there is a path with constant values of $\tilde{F}$ leading from y' to the global optimum $\hat{y}$. Thus y is connected by a $\tilde{F}$-non-increasing path to $\hat{y}$. Hence y is, by definition, not a strict local optimum. □

2.3 Adaptive Walks

An adaptive walk on a fitness landscape $(Y, \sim_Y, \tilde{F})$ is a Markov chain on the state space Y with transition probabilities $\pi_{y \to z} = 1/d_y$ for $y \sim_Y z$ and $\tilde{F}(z) \leq \tilde{F}(y)$. Otherwise $\pi_{y \to z} = 0$, except for $y = z$ where $\pi_{y \to y}$ is obtained by normalization of probability. The degree d_y of state y is the number of neighbours $|\{z \in Y : z \sim_Y y\}|$.

Call $\hat{Y}$ the set of global minima of the landscape $(Y, \sim_Y, \tilde{F})$. Assume that this landscape does not have a strict local minimum. Then the support of each invariant measure of the adaptive walk on this landscape lies in $\hat{Y}$. In the absence of strict local minima, each realization of an adaptive walk eventually hits a global minimum. Property (R2) clearly is a necessary condition for an optimization problem to be solvable by adaptive walks alone. The conditions of Theorem 1 are already sufficient as it excludes strict local optima.

Our aim in the following is therefore to construct cover-encoding maps that satisfy these conditions. We will see that this can indeed be achieved for several well-studied combinatorial optimization problems.

2.4 Examples of Cover-Encoding Maps

2.4.1 Square encoding for arbitrary landscapes

For any landscape $(X, \sim_X, f)$ we demonstrate existence of a cover-encoding map that gives rise to a landscape without strict local minima. We build ϕ on the product set $Y = X \times X$, so a local search can keep the best solution so far in one variable and use the other variable for exploration. We define $\phi : Y \to 2^X$ by setting

$$\phi((\xi,x)) = \{\xi, x\} \tag{3}$$

for all $(\xi, x) \in Y$.

Considering the properties of ϕ, (Y0) is obtained with $|\phi(y)| > 0$ for all $y \in Y$; (Y2) is fulfilled choosing $y = (x, x)$ for any $x \in X$. This implies (Y0) so ϕ is a cover-encoding map. We have (Y3) only in the trivial case $|X| \leq 2$. Property (F0) is fulfilled with $\hat{y} = (\hat{x}, \hat{x})$.

We define the neighbourhood relation $\sim_Y$ on Y by

$$(x_1, x_2) \sim_Y (\xi_1, \xi 2) \Leftrightarrow (x_1 = \xi_1 \wedge x_2 \sim_X \xi_2) \vee (x_2 = \xi_2 \wedge x_1 \sim_X \xi_1) \tag{4}$$

Thus the graph $(Y, \sim_Y)$ is the square of the graph $(X, \sim_X)$ in w.r.t. the Cartesian graph product. In particular, $(Y, \sim_Y)$ is connected because $(X, \sim_X)$ is connected. For $y, y' \in Y$, we write $d_Y(y, y')$ for the standard graph distance, the length of a shortest path, between y and y'; analogous notation for the distance d_X on $(X, \sim_X)$. For $(x_1, x_2) \in Y$ and $(\xi_1, \xi_2) \in Y$ we have $d_Y((x_1, x_2), (\xi_1, \xi_2)) = d_X(x_1, \xi_1) + d_X(x_2, \xi_2)$.

Now let $(x_1, x_2) = y \in Y \setminus \{(\hat{x}, \hat{x})\}$. Then $x_1 \neq \hat{x} \neq x_2$. We assume, without loss of generality, $f(x_1) \geq f(x_2)$ (otherwise swap x_1 and x_2). Because $(X, \sim_X)$ is connected, we find a neighbour $x' \sim_X x_1$ with $d_X(x', \hat{x}) = d_X(x_1, \hat{x}) - 1$. With $y' = (x', x_2)$, we have $\tilde{F}(y') = \min\{f(x'), f(x_2)\} \leq f(x_2) = \tilde{F}(y)$ and $d_Y(y', \hat{y}) = d_Y(y, \hat{y}) - 1$. For each element $y \in Y$ we thus find a $y' \in Y$ that (i) is strictly closer to $\hat{y}$ than y is; and (ii) does not evaluate at higher value than y under $\tilde{F}$. Using the argument inductively at most $d_Y(y, \hat{y})$ times, the desired sequences in (R1) and (R2) are constructed. Therefore properties (R1) and (R2) are fulfilled by $(Y, \sim_Y, \tilde{F})$. This landscape does not have strict local minima according to Theorem 1.

2.4.2 Prepartition Encoding for the NPP

An NPP instance is described by a list $(a_1, \ldots, a_n)$ of numbers. We write $[n] := \{1, \ldots, n\}$ for the index set. We have to devide these n numbers into two subsets with as equal a sum as possible. In other words, we assign to each index i a variable $x_i \in \{-1, +1\}$ so that

$$f(x) = \left| \sum_{i=1}^{n} x_i a_i \right| \to \min! \tag{5}$$

see e.g. (Mertens, 2006) for a review. The so-called *prepartitioning* encoding (Ruml et al., 1996) of the NPP can be interpreted as follows. Let $y : [n] \to [n]$ be an arbitrary function. It defines the partition $\Pi_y := \{y^{-1}(k) | 1 \leq k \leq n\}$ whose classes are the indices of the input numbers that are assigned the same value of y. As usual we write $[i]_{\Pi_y}$ for the class $y^{-1}(k)$ that contains index i. For given Π_y we now insist that the signs $x_i = x_j$ whenever $y(i) = y(j)$. This amounts to the restricted set of configurations

$$\phi(y) = \{x \in X | x_i = x_j \text{ if } j \in [i]_{\Pi_y}\}. \tag{6}$$

One easily checks that $\phi(y) = X$ whenever y is a bijection, i.e., (Y3) is satisfied. Furthermore, the subset $Y^* = \{y \in Y \big| |y([n])| = 2\}$ corresponds exactly to the assignments of positive and negative signs: Writing $y([n]) = \{p, q\}$ simply set $x_1 = +1$ if $y(i) = p$ and $x_1 = -1$ if $y(i) = q$. (More precisely, the choice of $x_1 = +1$ or $x_1 = -1$ is arbitrary; the symmetry can, however, easily be removed e.g. fixing $x_1 = +1$ once and for all.) Conversely, every assignment of signs has a representation as a bipartition in Y^*. Thus (Y2) is satisfied.

The most natural choice of an adjacency $\sim$ on Y is to define $y \sim y'$ if and only if $y(i) \neq y'(i)$ for exactly one $i \in [n]$. Unless y is a bijection, there is at least one unused value $k \in [n] \setminus y([n])$ and at least one pair $j', j'' \in [n]$ with $y(j') = y(j'')$. The neighbor y' of y with $y'(i) = y(i)$ for $i \neq j''$ and $y'(j'') = k$ corresponds to refinement of the partition Π_y because $[j']_{\Pi'_y} = [j']_{\Pi_y} \setminus \{y''\}$, $[j'']_{\Pi'_y} = \{j''\}$, and all other classes of Π'_y and Π'_y are the same. Thus $(Y, \sim)$ satisfies (R3).

An optimal solution $\hat{x}$ of the NPP (X, f) is a partition $\hat{\Omega}$ of $[n]$ into exactly two classes Q_+ and Q_- so that $x_i = +1$ for $i \in P_+$ and $x_i = -1$ for $i \in P_-$. A code $y \in Y$ is good if there is configuration in $\phi(y)$ in which the signs can be assigned in exactly this manner, i.e., if Π_y is a refinement of $\hat{\Omega}$. Conversely, if $\phi(y)$ is good only if it a refinement of a bipartition Ω that respresent a global minimum. Generically $\hat{\Omega}$ is unique. Now consider two classes P_1 and P_2 in Π_y that are contained in the small class of Ω, i.e., $P_1, P_2 \subset \Omega$. Reassigning one element at a time from P_2 to P_1 thus corresponds to a sequence of codes $y = y_1, y_2, \dots y_{|P_2|}$ all of which are encode refinements Ω. Furthermore, $y_{|P_2|}$ is one class less than y. Repeating this step at most $n - 2$ times eventually results in Ω. Intermediate codes y_i and y_{i-1} are adjacent by construction and satisfy $\tilde{F}(y_i) = F(\hat{y})$, i.e, condition (R1) is satisfied.

Thus we conclude that the "oracle landscape" $(Y, \sim, \tilde{F})$ has no strict local minima.

2.4.3 Prepartition Encoding for the TSP

The cost function of TSP (Gutin and Punnen, 2007) is

$$f(\pi) = \sum_{i=1}^{n} d_{\pi(i),\pi(i+1)} \tag{7}$$

where $\pi \in X$ is a bijection $\pi : [n] \to C$ from the index set $[n]$ to a set of cities C. The index i specifies the position along the tour. For a city c, therefore $\pi^{-1}(c)$ is its position along the tour. The problems is parametrized by distances $d : C \times C \to \mathbb{R}$ that satisfy $d(c, c) = 0$ for all $c \in C$ buy in general are neither symmetric nor do they satisfy the triangle inequality.

Klemm et al. (2012) introduced the following version of a prepartition encoding: An arbitrary function $y : C \to [n]$ is used to restrict the possible orderings of the cities along the tour by means of the following rule: For all cities $c, d \in C$ holds that $y(c) < y(d)$ implies $\pi^{-1}(c) < \pi^{-1}(d)$. Again this defines a subset X_y of the search space X of each y. We use the same definition of adjacency in Y. Here, constant functions y impose no restrictions on π, i.e, $\phi(y) = X$ whenever $y(i) = y(j)$ for all $i, j \in [n]$. On the other hand, if y is bijective then X_y consists only of a single tour since in this case $y(c) = \pi^{-1}(c)$ for all $c \in C$, i.e., $\pi = y^{-1}$. Thus (Y2) and (Y3) are satisfied.

To address properties (R2) and (R1) we first observe that given an encoding y we can always can move one city c with $y(c) = k$ to one of the classes defined by y with an adjacent values k'. More precisely, suppose k' is such that (a) there is a city d so that $y(d) = k'$ and there are no cities e with $y(e) = k''$ for any k'' between k and k'. If $k' > k$ the city that we can move is the one with $y(c) = k$ that appears last in the optimal tour $\omega \in \varphi(y)$; similarly, if $k' < k$, we can move the city c with $y(c) = k$ that appears first in the optimal tour $\omega \in \varphi(y)$. In the first case we can set $k < y'(c) \leq k'$, in the second case we can choose $k' \leq y'(c) < k'$. By construction $\omega \in \varphi(y')$, and therefore $\tilde{F}(y') \leq \tilde{F}(y)$. It is also clear from the construction that the step from y to y' can always be chosen to that the number of classes $|y^{-1}([n])|$ remains constant, increases by one $|y^{-1}([n])|$, or decreases by one – unless we already have $|y^{-1}([n])| = n$, in which case only a decrease is possible, or we have $|y^{-1}([n])| = 1$, in which case only an increase is possible. Thus we can always find a path along which $\tilde{F}(y')$ does increase and along which $|y^{-1}([n])|$ is non-increasing or non-decreasing, resp. Note the moves keeping $|y^{-1}([n])|$ constant might be necessary to stepwisely move the values $y(c)$ around in $[n]$ to have enough "space" to break up individual classes of y^{-1} so that its members in the end have consecutive values of y. It is not hard to convince oneself that this is always possible. As a consequence we can always connect any y to a code with a single class (for which $\phi(y) = X$). For two adjacent classes we simply join, one-by-one, the cities of the smaller class to the larger one. Furthermore, the single-class code can be broken by pulling a city at a time so that (R1) also holds. Note that (R3) is not necessarily satisfied, however.

In contrast to the previous example of the NPP, here the paths are much more involved and often longer. We therefore conjecture that prepartition encoding is less efficient for the TSP than for the NPP.

2.4.4 Spanning Forest Encoding for the NPP

A very different encoding for the NPP can be constructed as follows. Denote by Y the set of all spanning forests of the complete graph K_n. For a detailed discussion of the combinatorical of spanning forests we refer to (Teranishi, 2005). Since each connected component is a tree y_a is bipartite, there is a uniquely defined bipartition $(V^+_{y_a}, V^-_{y_a})$ of its vertex set. We assign $q_i = +1$ for $i \in V^+_{y_a}$ and $q_i = -$ for $i \in V^-_{y_a}$ to the other.

$$\phi(y) = \{x | x_i = p_a q_i,\, i \in V_{y_a},\, p_a = \pm 1\} \tag{8}$$

Suppose the spanning forest y has k components. Then the sign pattern on each component y_a is uniquely defined by fixing independently the sign of the lexicographically smallest $i \in V_{y_a}$. Thus $\phi(y)$ consists of exactly 2^k distinct configurations. It follows that $\phi(y) = X$ if y contains no edges and $\phi(y) = \{x, \bar{x}\}$ whenever y is a spanning tree. Since x and $\bar{x}$ represents the same solution of the number partitioning problem, φ satisfies (Y2) and (Y3).

(R3) holds since removing an edge from the spanning forrest y yields another spanning forrest y' that imposes fewer restrictions and thus corresponds to a larger subset of X. In general, write $y' \prec y$ if y' is a subforrest of y. Then $\varphi(y) \subset \varphi(y')$. The unconstrained search space corresponds to the spanning forrest y_0 without edges. Conversely, every spanning tree $\hat{t}$ that defines the bipartition of the globally minimal solution of the original NPP encodes exactly this solution. Every sequence $\hat{t} = y_{n-1} \succ y_{n-2} \succ \cdots \succ y_1 \succ y_0$ of spanning forrest obtained by succesive edge deletions from $\hat{t}$ connects y_0 and $\hat{t}$ and each $\phi(y_i)$ also contains the global minimum encoded by $\hat{t}$. Thus (R1) holds.

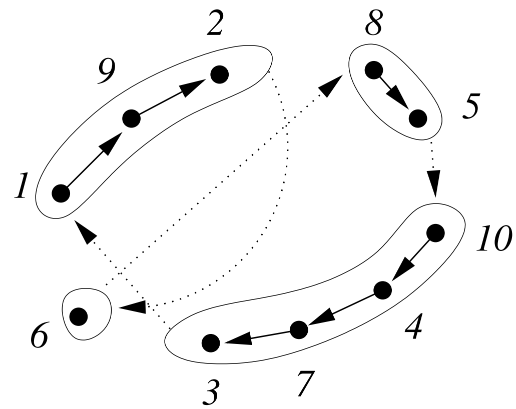

Figure 1: Example for a subdivision of the TSP. The cities are subdivided into classes of a partition within which their order is fixed among all restricted tours (full arrows). The order in which the classes are traversed remains free (dotted arrows).

2.4.5 Subdivision Encoding for the TSP

An alternative encoding for the TSP uses a permutation $\psi : [n] \to C$ of the set C cities and subdivision Π if $[n]$ into consecutive intervals. We specify Π by the upper bound of the interval, i.e., $I_u := \{k | i_{u-1} < k \le i_u\}$. Since the tours are circular, we set $i_0 = i_m$ and as usual consider the order $<$ circular on $[n]$. Therefore $I_1 := \{i_{m+1}, \dots, i_n, 1, \dots i_1\}$. An encoded configuation $y := (\psi, \Pi)$ fixes the order ψ of cities $\psi(k)$ within each of the index invervals I_u. The first city in interval I_u is $\psi(i_{u-1}+1)$, the last city is $\psi(i_u)$. Thus $\pi \in \phi(y)$ if π is obtained by permuting the intervals I_u and following the order given ψ within each interval.

If Π is the discrete partition, then we obviously have $\phi(y) = X$, which the indiscrete partition uniquely specifies the tour ψ. The encoding therefore satisfied (F0), (Y0), (Y1), (Y2), and (Y3). Consider any adjacency relation $\sim$ on Y so that $y \sim y'$ if Π' is obtained by splitting a class (interval) into two or merging two intervals. Then (R3) is clearly satisfied. It remains to consider (R1). To this end, we have elaborate on the definition $\sim$: we further require that $y \sim y'$ whenever $\Pi = \Pi'$ and ψ and ψ' are such that they describe the same order within intervals by specify a different order among them. This could be restricted to any ergodic move set on the permutations of intervals. Now consider an encoded configuration $\hat{y}$ with $\hat{x} \in \phi(\hat{y})$. The intervals of specified $\hat{y}$ are partial tours of the globally optimal solution. Moves on Y can now be performed so that a new encoding y' is stepwisely obtained that uses the same intervals and brings two partial tours that are consecutive in $\hat{x}$ into the desired order. During this stepwise change of ψ the encoded sets $\phi(y)$ stay the same, and thus $\phi(y') = \phi(\hat{y})$. Now the two appropriate consecutive intervals can be merged. This reduces m by 1 and makes $\phi(y)$ smaller, but the globally optimal solution is still retained, i.e., $\hat{x} \in \phi(y)$. The procedure can be repeated at most $m-1$ times to reach the indiscrete partition, which fully specifies the globally optimal tour. Thus (R1) holds for all choices of neighborhoods that allow merging/splitting of adjacent intervals and an ergoding permutation of the intervals.

2.4.6 Sparse subgraph encoding for the Maximum Matching Problem

For a graph $G = (V, E)$, a matching is a subset $M \subseteq E$ of pairwise disjoint edges, i.e. (V, M) is a graph with maximum degree at most 1. Denoting by X the set of matchings on G, the maximum matching problem (MMP) (X, f) has the cost function f giving the number of unmatched nodes

$$f(M) = |V| \setminus |\bigcup_{e \in M} e| \tag{9}$$

in a matching M. Thus the MMP asks for a subset of edges that cover as many nodes as possible without having any node contained in more than one edge (Lovász and Plummer, 1986).

Now consider an edge subset $S \subseteq E$. In the present context, we call S sparse if the graph (V, S) has maximum degree 2, so each connected component of (V, S) is a cycle or path (including isolated nodes as trivial paths). Denote by Y the set of all sparse subsets of E. Since a matching M is also a sparse subset of G, we have $X \subseteq Y$.

The cover-encoding map $\phi : Y \to 2^X$ assigns each $S \in Y$ the set of maximum matchings of the graph (V, S). Now with S sparse, the maximum matching problem on (V, S) is trivially solved separately on each connected component being a path or cycle. For a path of odd length k, the maximum matching is unique with $(k + 1)/2$ edges; a path or cycle of even length k has exactly two disjoint maximum mathchings of cardinality $k/2$. A cycle of odd length k has exactly k pairwise different maximal matchings of cardinality $(k - 1)/2$.

For each matching $x \in X$, we have $\phi(x) = \{x\}$ so property (Y2) holds. Properties (Y0) and (Y1) are fulfilled. With the choice $\hat{y} = \hat{x}$, (F0) is fulfilled. Property (Y3) holds if and only if (G, E) is sparse itself.

We consider sparse subsets D and D' as adjacent, $D \sim D'$, if they differ at exactly one edge, $|(L \cap L') \setminus (L \cup L')| = 1$.

For demonstrating properties (R1) and (R2), let $y \in Y \setminus \{\hat{y}\}$. We show that there is $y' \sim_Y y$ with $\tilde{F}(y') \leq \tilde{F}(y)$ and $|(y' \cup \hat{y}) \setminus (y' \cap \hat{y})| \leq |(y \cup \hat{y}) \setminus (y' \cap \hat{y})|$. Thus neighbour y' is obtained from y either by adding an edge contained in $\hat{y}$ or removing an edge not contained in $\hat{y}$. If $y \supset \hat{x}$, find an edge $e \in y \setminus \hat{x}$ and set $y' = y \setminus \{e\}$, done. Otherwise, since $y \neq \hat{y}$, there is an edge $\{v, w\} = e \in \hat{x} \setminus y$. If $y \cup \{e\} =: z$ is sparse, we are done using $y' = z$. Otherwise at least one of nodes v and w has degree 3 in the graph (V, z); suppose node v has degree 3. Find a maximum matching $x \in \phi(y)$. Since v has degree 2 in the graph (V, y), there is an edge $e' \in y \setminus x$ incident in v. Set $y' = y \setminus \{e'\}$. We easily confirm $\tilde{F}(y') \leq \tilde{F}(y)$ in each of the cases above. Sequences for properties (R1) and (R2) are obtained by induction.

2.4.7 String encoding for maximum clique

For a graph $G = (V, E)$, a clique is a node subset $C \subseteq V$ inducing a fully connected subgraph, i.e. $\{v, w\} \in E$ for all $v, w \in C$ with $v \neq w$. Denoting by X the set of cliques of G, the maximum clique problem (MCP) (X, f) has the cost function f giving the number of nodes

$$f(M) = |V \setminus C| \tag{10}$$

outside a clique M (Bomze et al., 1999).

For arbitrary $l \in \mathbb{N}$ and any string of not necessarily distinct nodes $(v_1, v_2, \ldots, v_l) \in$

V^l, we define the greedy clique $\gamma_G(v_1, v_2, \ldots, v_l) \subseteq V$ recursively by

$$\gamma_G(v_1, v_2, \ldots, v_l) = \begin{cases} \gamma_G(v_1, v_2, \ldots, v_{l-1}) \cup \{v_l\} & \text{if } \{v_i, v_l\} \in E \text{ for all } i \in [l-1] \\ \gamma_G(v_1, v_2, \ldots, v_{l-1}) & \text{otherwise} \end{cases} \tag{11}$$

and $\gamma_G(\varnothing) = \emptyset$ for the empty string $\varnothing$.

We construct a cover-encoding map ϕ based on strings of length $|V| =: n$, so $Y = V^n$. For a string $y \in Y$, we denote the substring (suffix) from index k to the end (index n) by $(y)_{k\dashv}$. Now ϕ maps a string $y \in Y$ to maximal greedy cliques over suffices of y,

$$\phi(y) = \{\gamma_G((y)_{k\dashv}) : k \in [n] \text{ and } \forall i \in [n] : \gamma_G((y)_{k\dashv}) \not\subset \gamma_G((y)_{i\dashv})\} \,. \tag{12}$$

So a clique C is contained in $\phi(y)$ if and only if C is a greedy clique from a suffix of y and none of the other greedy cliques from y properly contains C. This ensures that ϕ produces all the singletons, thus fulfilling property (Y2). We call y pure if $|\phi(y)| = 1$. A string $y \in Y$ is pure if and only if $\{y_i : i \in [n]\}$ is a clique of G. We define strings $y, y' \in Y$ to be adjacent, in symbols $y \sim_Y y'$, if and only if there is a unique index $i \in [n]$ with $y_i \neq y'_i$ (Hamming distance 1).

For proving properties (R1) and (R2), we first observe that there is a non-increasing sequence of strings from any $y \in Y$ to a pure $y^{(\mathrm{p})} \in Y$ with $\phi(y^{(\mathrm{p})}) \subseteq \phi(y)$ and $\tilde{F}(y^{(\mathrm{p})}) = \tilde{F}(y)$. The sequence is obtained by finding a maximal $C \in \phi(y)$. If y is not pure, there is $i \in [n]$ with $y_i \notin C$. The next string in the sequence can be obtained by replacing the entry y_i with an arbitrary element from C.

If $y, z \in Y$ are pure with $\phi(y) = \phi(z) = \{C\}$ and $|C| < n$, there is a non-increasing sequence from y to z. It may be constructed by stepwise swapping operations. Since $|C| < n$, there is at least one element in C found at two distinct positions in y so one of these can be used as a temporary variable in the swap.

Now let $y, y' \in Y$ with $\tilde{F}(y') \leq \tilde{F}(y)$. Find a maximal clique $C \in \phi(y)$ and a maximal clique $C' \in \phi(y')$. We construct a non-increasing sequence from y to y' by concatenating the following sequences. First, a non-increasing sequence from y to a pure $y^{(\mathrm{p})} \in Y$ with $\tilde{F}(y^{(\mathrm{p})}) = \tilde{F}(y)$. Second, a non-increasing sequence from $y^{(\mathrm{p})}$ to a pure $z \in Y$ with $\{z_1, z_2, \ldots, z_{|C|}\} = C$ and $\{z_1, z_2, \ldots, z_{|C \setminus C'|}\} = C \setminus C'$, and arbitrary $z_{|C|+1}, z_{|C|+2}, \ldots, z_n \in C$. Third, a sequence from z to a string z' is obtained by assigning, step by step, nodes in $C' \setminus C$ to entries from $z_{|C|+1}$ to z_n. The sequence is non-increasing because each ofits strings generates C under ϕ. On the other hand, $\gamma_G((z')_{(|C \setminus C'|+1)\dashv}) = C'$ so $\tilde{F}(z') = \tilde{F}(y')$. Now again by swap steps, we transform z' into y'.

3 Coarse Graining

Some of the restricted search spaces $\phi(y)$ introduced above can also be thought of as coarse grainings of the original problem. This is quite obvious in the case of the prepartition encoding of the NPP.

3.1 Prepartition Encoding of the NPP

Consider the NPP instance with numbers $\{a_1, a_2, \ldots, n\}$ and let $\Pi = \{P_1, \ldots, P_m\}$ be an arbitrary partition of $[n]$ with classes (subsets) P_j. Of course, we can think of Π as the classes defined by the prepartition encoding, i.e. $\Pi = \{y^{-1}(k) | k \in [n]\}$. Of course $m \leq n$. Set $b_j = \sum_{i \in P_j} a_i$. Then the set of numbers $\{b_1, \ldots, b_m\}$ defines an NPP on m numbers. In terms of a prepartition y this amounts to $b_k = \sum_{i \in y^{-1}(k)} a_i$. Of course, if

$m = n$, then Π is the discrete partition in which every class P_j contains only a single element, and hence $\{a_1, \dots a_n\} = \{b_1, \dots, b_m\}$. In the general case the solutions of the two NPPs are related to each other in the following way. Denote the variables for the smaller NPP by $x'_j \in \{+1, -1\}$ and write f_a and f_b for the cost functions. Then, obviously

$$f_a(x) = f_b(x') \text{ whenever } x_i = x'_j \text{ for all } i \in P_j \tag{13}$$

An optimal solution $\hat{x}$ of the larger problem (X, f_a) corresponds to a partition $\hat{\Omega}$ of $[n]$ into exactly two classes Q_+ and Q_- so that $x_i = +1$ for $i \in P_+$ and $x_i = -1$ for $i \in P_-$. The coarse grained NPP (X', f_b) has an optimal solution with the same cost if (and in the generic case also only if) $P_j \subseteq P_+$ or $Q_j \subseteq Q_-$ holds for all $j \in [m]$, i.e., if (and generically only if) the coarse graining partition Π is a refinement of the partition $\hat{\Omega}$ that encodes the globally optimal solution of the original problem.

3.2 Coarse Graining and the Renormalization Group

A very general formal approach to analyzing spin glasses and related disordered systems is "renormalization" (Rosten, 2012). In our context it can be described as follows. For given type of problems, such as the NPP or the TSP, consider the space $\mathfrak{X}$ of all possible instances of all sizes. A particular instance (e.g. the NPP with n numbers $a = \{a_1, a_2, \dots, a_n\}$) is a point $\mathbf{x} \in \mathfrak{X}$. Now we define a set $\mathcal{R}$ of maps $r : \mathfrak{X} \to \mathfrak{X}$ that map larger instances to strictly smaller ones. Of interest in the context are in particular those maps r that (approximately) preserve salient properties. Since $r(\mathbf{x})$ is a smaller instance than $\mathbf{x}$, the map r is not invertible. The maps in $\mathcal{R}$ can of course be composed, and thus form a semi-group which is known as the *renormalization group* (Wilson and Kogut, 1974; Wilson, 1971).

In spin glass physics, $\mathcal{R}$ is chosen e.g. as so-called block spin transformations (Kadanoff, 1966), i.e., suitable average over small local subsets of spins. The coarse graining in the previous subsection is an example. However, only certain block transformations, i.e., sums of numbers a_i, preserve the optimal solutions. In statistical physics, one is therefore primarily interested in the typical behavior of block transformations acting on probability distributions. Here, however, our focus is on particular instances. Starting from $\mathbf{x} = (X, f)$, or more precisely, an encoding y so that $\phi(y) = \mathbf{x}$, we can think adjacent encodings $y' \sim y$ with $|\phi(y')| < |\phi(y)|$ as "renormalized" versions of $\phi(y)$. A path in $(Y, \sim)$ leading from $\mathbf{x}$ to the trivial instance thus can be seen as the iteration of renormalizations.

3.3 Travelling Salesman Problems

Recall the subdivision encoding for the TSP and fix an encoding $y = (\psi, \Pi)$. The length of the partial tour inside inverval I_u is

$$\ell_u = \sum_{k=i_{u-1}+2}^{i_u} d_{\psi(k-1)\psi(k)} \tag{14}$$

Furthermore, the road from interval I_p to interval I_q is road from $\psi(i_p)$ to $\psi(i_{q-1}+1)$, i.e.,

$$\tilde{d}_{pq} = d_{\psi(i_p),\psi(i_{q-1}+1)} \tag{15}$$

Since a tour $\pi \in \phi(y)$ is uniquely defined by a permutation $\xi : [m] \to [m]$ of the intervals, we have

$$\ell(\pi) = \tilde{\ell}(\xi) + \sum_{u=1}^{m} \ell_u \tag{16}$$

where $\tilde{\ell}(\xi) = \sum_i \tilde{d}_{\xi(i),\xi(i+1)}$ is the tour length of the TSP restricted to the connections between the fixed intervals. With a slight change one can also produce a TSP that retains the original values of the cost function. To this end we set

$$d'_{pq} = d_{\psi(i_p),\psi(i_{q-1}+1)} + (\ell_p + \ell_q)/2 \tag{17}$$

and $\ell'(\xi) := \sum_i \tilde{d}'_{\xi(i),\xi(i+1)}$. A short computation verifies $\ell(\pi) = \ell'(\xi)$.

Note that we naturally obtain an asymmetric TSP even if the original problem was symmetric since now $d'_{pq} \neq d'_{qp}$ because in general we will have $d_{\pi(i_p)\pi(i_{q-1}+1)} \neq d_{\pi(i_q)\pi(i_{p-1}+1)}$.

3.4 Spanning Forest Representation of the NPP

Let us now return to the NPP. Let y be a spanning forrest of K_n. For each connected component (tree) $t\dot{\subseteq}y$ let V_t^+ and V_t^- be the corresponding bipartition of the vertex set of t. Define

$$b_t = \left| \sum_{i\in V_t^+} a_i - \sum_{i\in V_t^-} a_i \right| \tag{18}$$

This defined a instance of the NPP with as many numbers b_t as connected components in y. A choice of sign $z_t \in \{+1,-1\}$ for t implies a particular choice of sign for each a_i, i.e., each configuration z for the NPP with numbers $\{b\}$ corresponds to a configuration x of the original problem with numbers $\{a\}$. Clearly, these coincide with the configurations $\phi(y)$ described in Sect. 2.4.4.

3.5 Not all Restrictions are proper Coarse Grainings

The prepartition encoding of the TSP, on the oder hand, cannot be rephrased as a coarse grained (i.e., reduced-size) TSP. To see this, simply observe that the evaluation of a tour in the restricted model still requires an optimization over multiple incoming and outgoing connections (roads) for every city, i.e., the information of inter-city distances cannot by collapsed in any way upon the transition from a larger (less restricted) to a smaller (more restricted) problem. Nevertheless, it makes sense to consider cover-encoding maps ϕ that do not have a direct interpretation as a coarse graining or a renormalization.

4 Heuristic optimization over Y

So far, we were only concerned with the abstract structure of cover-encoding maps $\phi : Y \to 2^X$ and the adjacencies $\sim$ in their encodings Y. On this theoretical basis, we can now construct a search-based *optimization heuristic* that generalized the approaches in (Ruml et al., 1996) and our earlier work (Klemm et al., 2012). The idea is very simple: If we have an accurate and efficiently computable heuristic, we can quickly obtain good upper bounds $\alpha_f(y) \geq \tilde{F}(y)$ for each of the restricted problems $(\phi(y), f)$. The properties (R1) and (R2) guarantee the existence of non-increasing paths from an arbitary initial encoding y_0 down to a final encoding $\hat{y}$. Steps to adjacent encodings that decrease α_f therefore will have a bias towards the optimal solution of the original problem.

The fact that we have to rely on the quality of the estimate $\alpha_f(y) \approx \tilde{F}(y)$ also suggests that it should be more efficient to restart the search often rather than try to overcome barriers of local minima in the landscape (Y, α_f). In the examples above local minima in (Y, α_f) can, as we have proved, appear only due to insufficient accuracy of the heuristic solutions $\alpha_f(y)$ for some encodings.

The discussion above also implies guidelines for the construction of encodings:

1. The cover-encoding map $\phi : Y \to X$ should be of a form that guarantees that $(Y, \sim, \tilde{F})$ has no local optima, i.e., the properties (R1), (R2), (Y1), and (Y2) should hold.

2. The paths in $(Y, \sim)$ connecting large sets $\phi(y)$ to smaller ones should not contain many steps along which the sets do not shrink. For instance, while the prepartition encoding for the NPP always has a strictly coarse-grained neighbor, this is not the case for the prepartition encoding for the TSP. We therefore suspect that other encodings for the TSP will work better in general.

3. The heuristic producing $\alpha_f(y)$ needs to be efficient, ideally not much slower than the function evalutions for the initial cost function f.

5 Numerical Experiments

In order to demonstrate that the theory developed above may also have practical implications we probe instances of encoded landscapes by adaptive walks. To simulate a realization of an adaptive walk, we first generate an initial state $y(0)$ by a procedure specific for the give landscape. At each time step t, we uniformly draw a neighbour z of state $y(t)$ and set $y(t+1) = z$ if $\tilde{F}(z) \leq \tilde{F}(y(t+1))$, $y(t+1) = y(t)$ otherwise.

We select the MMP and the MCP as examples because (1) oracle functions and encodings can constructed that guarantee the absence of strict local minima; and (2) there is a simple and efficient algorithm for exact computation of $\tilde{F}(y)$ for each $y \in Y$. So we do not require heuristics. We leave the combination of cover encoding maps with non-trivial heuristics for a future manuscript.

5.1 Maximum matching

Figure 2 shows the time evolution of cost in adaptive walks on the encoded landscapes of matchings encoded by sparse graphs. See the figure caption for details on the instances and section 2.4.6 for definitions. Note the logarithmic time axis in the plot.

Both on purely random graphs and on those with a planted perfect matching, a solution of globally minimal cost is found. In addition to reaching a minimum cost solution, we observe another interesting feature of the dynamics. The sizes of symbols (and annotated values in the uppermost curve) indicate the number of degrees of freedom $\delta = \log_2 |\phi(y(t))|$ of the solution $y(t)$ at time t. This is the number of the sparse graph's connected components with two distinct maximum matchings. Departing from a singleton state ($\delta = 0$), the number of degrees of freedom first increases and then decreases during the descent of cost. So the optimization happens as a walk through states $y \in Y$ with large cardinality $|\phi(y)|$ of the encoded set. Furthermore as a particular feature of this encoded landscape, the optimization dynamics eventually returns to low δ, having $|\phi(y(t))| = 1$ with a single optimal solution selected at large time t.

5.2 Maximum clique

Figure 2 shows the time evolution of the cost of adaptive walks on the encoded landscapes of graph cliques encoded by node sequences. See the figure caption for details on the instances and section 2.4.7 for definitions. We plot the difference with the minimum cost $\tilde{F}(y)$. A plotted value of 0 means the global optimum has been found.

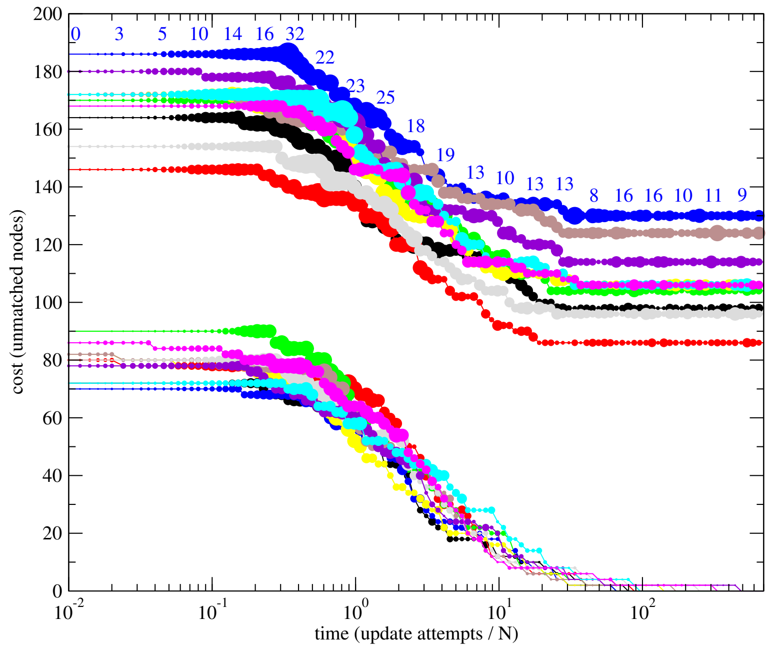

Figure 2: Time evolution of cost in adaptive walks on the landscape of matchings encoded by sparse subgraphs. Radius of symbols is proportional to the number of degrees of freedom (paths of even length $\neq 0$ and cycles of odd length) in the encoded state. Upper set of curves: 10 realizations, each on an independently generated ER random graph on 500 nodes with edge probability $p = 2/(N-1)$, i.e. average degree 2. Lower set of curves: 10 realizations on graphs (500 nodes) with perfect matching planted first, then adding each of the remaining possible edges with $p = 1/(N-2)$, resulting in average degree 2. Each adaptive walk is initialized by a random maximal matching $L(0)$. Departing from empty set, $L(0)$ is generated by considering the edges of the graph G in the order of a random uniform permutation and adding an edge to $L(0)$ if the result remains a matching.

As far as we can conclude from this preliminary analysis, time to reach the optimal solution scales moderately with problem size. Standard deviation over realizations (error bars in the plot) indicates also moderate variation of optimization time across these randomly generated instances.

6 Discussion and Conclusions

In this contribution we have shown that, in principle, it is possible to construct a genotypic encoding for any given phenotypically encoded combinatorial optimization problem with the property that the encoded landscape has no strict local minima. The construction hinges on three ingredients: a cover-encoding map $\phi : Y \to 2^X$ that satisfies a few additional conditions, a suitable adjacency relation on Y, and an oracle function that (miraculously) returns the optimal cost value on the restrictions of the original problem to the covering sets $\phi(y)$. Of course, if we had such an oracle function in practice, we would not need a search heuristic in the first place.

Nevertheless, the theory developed here is not just an empty excercise. It was ac-

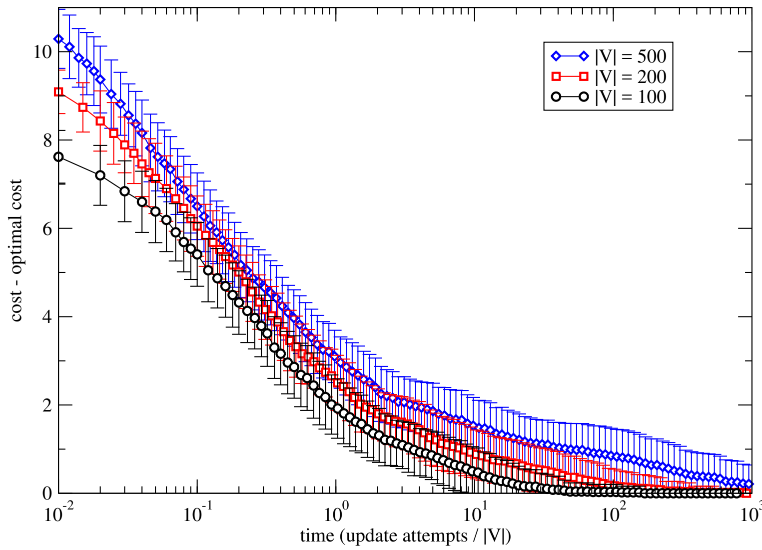

Figure 3: Time evolution of cost in adaptive walks on the landscape of cliques encoded by node sequences. For each graph size $|V|$, 100 random graph instances with parameter $p = 1/2$ are generated independently. For each instance, an adaptive walk on the encoded landscape is performed with starting state $(1, 1, \ldots, 1)$. Plotted values are difference between $\tilde{F}(y(t))$ of the state $y(t)$ held by the adaptive walk at time t and the optimal cost $F(\hat{x})$, averaged over the 100 instances. Length of error bars is the standard deviation over these instances. The exact $F(\hat{x})$ is computed with a branch-and-bound algorithm (Östergård, 2002).

tually motivated by the fact that the prepartition encoding proposed by Ruml et al. (1996) yields excellent solutions for the NPP and the desire to understand (a) why this particular method works so well and (b) how it can be generalized to essentially arbitrary combinatorial optimization problems in a principled way. The key to both question was the observation that the fitness evaluation for a prepartition with the help of the Karmarkar-Karp differencing algorithm (Karmarkar and Karp, 1982; Boettcher and Mertens, 2008) provides a very good approximation to the oracle function for the NPP. Thus in fact, it is perfectly resonable to investigate the idealized case in which the oracle function is known exactly.

The numerical simulations of Section 5 strongly suggest that encodings with local-minima-free landscapes indeed admit efficient optimization by local search based methods. Hence the theoretical results obtained here are likely to be of practical relevance provided a sufficiently accurate approximation to the oracle function can be computed. It remains an open question for future research what it means for an approximation to be sufficiently accurate. We suspect, however, that the main problem arises when the approximation claims $\alpha_f(y') < \alpha(y)$, suggesting to accept a step from y to y', while $\tilde{F}(y') > \tilde{F}(y)$ and hence the step to y' should not have been taken.

The construction of encoding for several well-known optimization problems also highlighted connections encoding and a natural notion of coarse-graining for optimization problems. This also suggests a link to renormalization group methods commonly used in physics. While it is clear that there is not a 1–1 correspondence, and coarse-grainings are just a particular subclass of encodings, this connection certainly deserves further study. The formalism laid out here at least provides a promising starting point.

Acknowledgements

KK acknowledges funding from MINECO through the Ramón y Cajal program and through project SPASIMM, FIS2016-80067-P (AEI/FEDER, EU). This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement N. 694925).

References

Applegate, D. L., Bixby, R. M., Chvátal, V., and Cook, W. J. (2006). *The Traveling Salesman Problem*. Princeton Univ. Press.

Banzhaf, W. and Leier, A. (2006). Evolution on neutral networks in genetic programming. In Yu, T., Riolo, R., and Worzel, B., editors, *Genetic Programming Theory and Practice III*, pages 207–221. Springer, New York.

Boettcher, S. and Mertens, S. (2008). Analysis of the Karmarkar-Karp differencing algorithm. *Eur. Phys. J. B*, 65:131–140.

Bomze, I. M., Budinich, M., Pardalos, P. M., and Pelillo, M. (1999). The maximum clique problem. In Du, D.-Z. and Pardalos, P. M., editors, *Handbook of Combinatorial Optimization – Supplement Volume A*, pages 1–74. Kluwer Academic Publishers, Dordrecht.

Borenstein, Y. and Moraglio, A., editors (2014). *Theory and Principled Methods for Designing Metaheuristics*. Springer-Verlag, Berlin, Heidelberg.

Choi, S.-S. and Moon, B.-R. (2008). Normalization for genetic algorithms with nonsynonymously redundant encodings. *IEEE Trans. Evol. Comp.*, 12:604–616.

Engelbrecht, A. and Richter, H., editors (2014). *Recent advances in the theory and application of fitness landscapes*. Springer-Verlag, Berlin, Heidelberg.

Fernández, P. and Solé, R. V. (2007). Neutral fitness landscapes in signalling networks. *J. R. Soc. Interface*, 4:41–47.

Flamm, C., Hofacker, I., Stadler, B., and Stadler, P. (2007). Saddles and barrier in landscapes of generalized search operators. *LNCS*, 4436:194–212.

Gutin, G. and Punnen, A. P., editors (2007). *The Traveling Salesman Problem and Its Variations*, volume 12 of *Combinatorial Optimization*. Springer-Verlag, Berlin, Heidelberg.

Kadanoff, L. P. (1966). Scaling laws for ising models near t_c. *Physics*, 2:263–272.

Karmarkar, N. and Karp, R. M. (1982). *The differencing method of set partitioning*. Computer Science Division (EECS) University of California, Berkeley, CA.

Klemm, K., Mehta, A., and Stadler, P. F. (2012). Landscape encodings enhance optimization. *PLoS ONE*, 7:e34780.

Knowles, J. D. and Watson, R. A. (2002). On the utility of redundant encodings in mutation-based evolutionary search. *LNCS*, 2439:88–98.

Lovász, L. and Plummer, M. D. (1986). *Matching Theory*, volume 29 of *Annals of Discrete Mathematics*. North-Holland, Amsterdam, NL.

Mertens, S. (2006). The easiest hard problem: Number partitioning. In Percus, A., Istrate, G., and Moore, C., editors, *Computational complexity and statistical physics*, pages 125–140. Oxford University Press, Oxford.

Neumann, F. and Witt, C. (2010). *Bioinspired Computation in Combinatorial Optimization*. Springer-Verlag, Berlin, Heidelberg.

Östergård, P. R. J. (2002). A fast algorithm for the maximum clique problem. *Discr Appl Math*, 120:197–207.

Østman, B., Hintze, A., and Adami, C. (2010). Critical properties of complex fitness landscapes. In Fellermann, H., Dörr, M., Hanczyc, M. M., Laursen, L. L., Maurer, S. E., Merkle, D., Monnard, P.-A., Støy, K., and Rasmussen, S., editors, *Artificial Life XII*, pages 126–132. MIT Press.

Reidys, C. M. and Stadler, P. F. (2002). Combinatorial landscapes. *SIAM Review*, 44:3–54.

Rosten, O. J. (2012). Fundamentals of the exact renormalization group. *Physics Reports*, 511:177–272.

Rothlauf, F. (2006). *Representations for Genetic and Evolutionary Algorithms*. Springer, Heidelberg, 2nd edition.

Rothlauf, F. (2011). *Design of Modern Heuristics: Principles and Application*. Springer-Verlag.

Rothlauf, F. and Goldberg, D. E. (2003). Redundant representations in evolutionary computation. *Evol. Comp.*, 11:381–415.

Ruml, W., Ngo, J., Marks, J., and Shieber, S. (1996). Easily searched encodings for number partitioning. *J. Opt. Th. Appl.*, 89:251–291.

Schuster, P., Fontana, W., Stadler, P. F., and Hofacker, I. L. (1994). From sequences to shapes and back: A case study in RNA secondary structures. *Proc. Roy. Soc. Lond. B*, 255:279–284.

Teranishi, Y. (2005). The number of spanning forests of a graph. *Discrete Math.*, 290:259–267.

Wilson, K. G. (1971). Renormalization group and critical phenomena. i. renormalization group and the kadanoff scaling picture. *Phys. Rev. B*, 4:3174–3183.

Wilson, K. G. and Kogut, J. (1974). The renormalization group and the ϵ expansion. *Phys. Rep.*, 12:75–199.

Yu, T. and Miller, J. F. (2002). Finding needles in haystacks is not hard with neutrality. *Lect. Notes Comp. Sci.*, 2278:13–25. EuroGP 2002.