Barrier Trees on Poset-Valued Landscapes

Peter F. Stadler[a,b,c] and Christoph Flamm[a]
[a] *Institut für Theoretische Chemie und Molekulare Strukturbiologie*
Universität Wien, Währingerstrasse 17, A-1090 Wien, Austria
[b] *Bioinformatik, Institut für Informatik,*
Universität Leipzig, Kreuzstraße 7b, D-04103 Leipzig, Germany
[c] *Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe NM 87501, U.S.A.*
(`{studla,xtof}@tbi.univie.ac.at`*)*

12 Sep 2002

Abstract. Fitness landscapes have proved to be a valuable concept in evolutionary biology, combinatorial optimization, and the physics of disordered systems. Usually, a fitness landscape is considered as a mapping from a configuration space equipped with some notion of adjacency, nearness, distance, or accessibility, into the real numbers. In the context of multi-objective optimization problems this concept can be extended to poset-valued landscapes. In a geometric analysis of such a structure, local Pareto points take on the role of local minima. We show that the notion of saddle points, barriers, and basins can be extended to the poset-valued case in a meaningful way and describe an algorithm that efficiently extracts these features from an exhaustive enumeration of a given generalized landscape.

Table of Contents

1. Introduction

Solving optimization problems with multiple, often conflicting, objectives is in general a very difficult task. Evolutionary algorithms were adapted in the last decades to this generic problem class, see e.g. [22] for a recent review. In the case of single objectives a rather elaborate theory of "fitness" landscapes (for a review see [21]) has been developed, which so far has not been systematically extended to the multi-objective case. In this short contribution we concentrate on one aspect that has received very little attention so far, namely the saddle points connecting the basins associated with locally optimal solutions and the barriers in the cost function that separate locally optimal solutions from better ones. In contrast to much of the literature, which deals with a continuum setting, we restrict our attention to combinatorial multi-objective problems.

Consider a general multi-objective combinatorial minimization problem [7]. The "decision space" thus consists of a finite set X of "configurations". Furthermore, we are given a finite number K of objective functions $f_i : X \to \mathbb{R}$, $1 \le i \le K$. One says x weakly dominates[1] y, in symbols $x \preceq y$, if $f_i(x) \le f_i(y)$ for all i; x (strictly) dominates y, $x \prec y$ if $x \preceq y$ and there is at least one index i such that $f_i(x) < f_i(y)$. A point $x \in X$ is *Pareto minimal* if it is not strictly dominated by any other point. The task of multi-objective minimization is therefore to identify all Pareto minima, see e.g. [6]. Algorithms for finding Pareto minimal points exhaustively or by means of GAs are described e.g. in [3, 12, 15].

Multi-objective minimization may be seen as a generalization of conventional minimization in which the linearly ordered "cost space", typically the real numbers $\mathbb{R}$, is replaced by a partially ordered set, or *poset* for short, $(\mathbb{Y}, \preceq)$. Formally, a poset satisfies the following axioms for all $u, v, w \in \mathbb{Y}$

(P0) $u \preceq u$;

(P1) $u \preceq v$ and $v \preceq w$ implies $u \preceq w$;

(P2) $u \preceq v$ and $v \preceq u$ implies $u = v$;

Posets are more general than linearly ordered sets since we do not require that $u \preceq v$ or $v \preceq u$ for all $u, v \in \mathbb{Y}$. If neither of these two inequalities holds then u and v are *incomparable*, see e.g. [5]. The K objective functions $f_i : X \to \mathbb{R}$ can be viewed as a single objective function $f : X \to (\mathbb{R}^k, \le_i)$ where $f(x) = (f_1(x), \ldots, f_K(x))$ and the set $\mathbb{R}^k$ is endowed with the standard (componentwise) partial order, i.e., $f(x) \preceq f(y)$ iff $f_i(x) \le f_i(y)$ for all i, $1 \le i \le K$. For an arbitrary poset-valued cost function $f : X \to \mathbb{Y}$ we say that x weakly dominates y if $f(x) \preceq f(y)$ and x strictly dominates y if $f(x) \preceq f(y)$ and $f(x) \neq f(y)$, i.e., $f(x) \prec f(y)$.

[1] All inequalities are reversed in the case of maximization problems, of course.

In many cases multi-objective optimization is performed by means of heuristic search algorithms [4, 6, 7]. It is natural therefore to ask whether the theory of fitness landscapes can be extended to the poset-valued case. Formally, a landscape is triple $(X,\mathcal{X},f)$ consisting of set of configurations X, a topological structure $\mathcal{X}$ that determines the mutual accessibility of configurations and a cost or "fitness" function $f : X \to \mathbb{R}$. The neighborhood relation $\mathcal{X}$ is typically defined by the "move set" of a search heuristic. In the case of optimization algorithms it can be chosen by the user; in the case of biological applications it is typically defined by the mechanisms of mutation or recombination. In this contribution we will restrict ourselves to the simplest case in which the configuration space $(X,\mathcal{X})$ is a finite undirected graph $\mathbb{G} = (X,E)$ with vertex set X and edge set E. Here edges connect configurations that can be inter-converted by a single move.

The definition of a poset-valued combinatorial landscape is straight-forward: we simply replace the real-valued cost function $f : X \to \mathbb{R}$ by a poset-valued function $f : X \to \mathbb{R}$. In a classical landscape, a configuration x is a global minimum if $f(x) \leq f(y)$ for all $y \in X$ and a local minimum if $f(x) \leq f(y)$ for all neighbors y of x. The generalization to the poset-valued case is straight-forward and well known: A configuration x is a *Pareto minimum* if x is not strictly dominated by any configuration y, and a *local Pareto minimum* if it is not strictly dominated by a $\mathbb{G}$-neigboring configuration. Pareto minima therefore take the place of ordinary minima in a poset-valued landscape. This is in complete analogy to the better-studied continuum case [6] and provides a first hint that the notion of a poset-valued landscape might be useful.

The topic of this article is *not* optimization per se. Instead, we are interested in generalizing methods for characterizing geometrical properties of ordinary landscapes to the poset-valued case. In particular, we will be concerned with the saddle points that separate local Pareto points from each other and the barriers that separate local Pareto points from better solutions. To this end, we describe a method for efficiently extracting the relevant information from small toy problems. In fact, we assume throughout this contribution that we can enumerate the decision space X exhaustively.

2. Basins and Barriers

Let us first consider the case of a linearly ordered objective space, i.e., a conventional fitness landscape. In spin glass physics, for instance, the nontrivial breaking of ergodicity in spin glass systems is usually described by the so-called many-valley scenario in which the ease with which one valley can be reached from another one depends on the saddle points connecting them [20].

Similarly, basins and barriers play a crucial role for the folding dynamics of biopolymers [23, 9], and in the theory of Simulated Annealing [1, 2].

The energy of the lowest saddle point separating two local minima x and y is

$$\hat{f}[x,y] = \min_{\mathbf{p}\in\mathbb{P}_{xy}} \max_{z\in\mathbf{p}} f(z) \tag{1}$$

where $\mathbb{P}_{xy}$ is the set of all paths $\mathbf{p}$ connecting x and y by a series of consecutive mutations.

If the energy function is non-degenerate, i.e., two configurations have distinct fitness values, then there is a unique saddle point $s = s(x,y)$ connecting x and y characterized by $f(s) = \hat{f}[x,y]$. The extension to degenerate fitness functions is discussed in detail in [11]. To each saddle point s there is a unique collection of configurations $B(s)$ that can be reached from s by a path along which the energy never exceeds $f(s)$. In other words, the configurations in $B(s)$ are mutually connected by paths that never go higher than $f(s)$. This property warrants to call $B(s)$ the *valley or basin below the saddle* s. Furthermore, suppose that $f(s) < f(s')$. Then there are two possibilities: if $s \in B(s')$ then $B(s) \subseteq B(s')$, i.e., the basin of s is a "sub-basin" of $B(s')$, or $s \notin B(s')$ in which case $B(s) \cap B(s') = \emptyset$, i.e., the valleys are disjoint. This property arranges the local minima and the saddle points in a unique hierarchical structure which is conveniently represented as a tree, termed *barrier tree*.

In the poset case we have to modify the paths $\mathbf{p}$ that enter the definition in equ.(1). Adjacent configurations are not necessarily $\preceq$-comparable, hence we cannot determine an analogue of height along arbitrary path. As a consequence, we have to require that $\mathbb{P}_{xy}$ consists only of those paths in which *consecutive* configurations are comparable. Let $\mathbf{p} \in \mathbb{P}_{xy}$. Then

$$\sigma(\mathbf{p}) = \{x \in \mathbf{p} |\ \nexists y \in \mathbf{p} : f(x) \prec f(y)\} \tag{2}$$

consists of all points along a path that do not strictly dominate another point in $\mathbf{p}$. Obviously, $\sigma(\mathbf{p})$ plays the role of the local maxima along $\mathbf{p}$. In particular, $\sigma(\mathbf{p})$ contains the points in $\mathbf{p}$ with maximal values of any linear extension of $\prec$. Next we consider the union

$$\Sigma_{xy} = \bigcup_{\mathbf{p}\in\mathbb{P}_{xy}} \sigma(\mathbf{p}) \tag{3}$$

of the "maxima" along all possible paths. The set $S(x,y)$ of *poset-saddle points between x and y* can now be defined as the Pareto points in Σ_{xy}, i.e., as

$$S(x,y) = \left\{z \in \Sigma_{x,y} \middle|\ \nexists u \in \Sigma_{x,y} : f(u) \prec f(z)\right\} \tag{4}$$

In particular, $S(x,y)$ contains the saddle points w.r.t. all linear extensions of the partial order. It should be noted that the poset-saddle points defined here are completely different from the saddle points of the Lagrangian-type scalarizations of multi-objective optimization problems that are considered e.g. in [14].

The direct evaluation of equ.(1) is an extremely laborious task even in ordinary landscapes [19] that becomes hopeless already for moderately small toy landscapes. Consequently, do not attempt to use eqns. (2–4). Instead, we follow the approach of [9] and modify the "flooding algorithm" `barriers`. This approach, however, forces us to be content with one representative of $S(x,y)$ for each pair of local Pareto points x and y.

3. The Flooding Algorithm

Let us first consider the case of linear order [9, 11]. Starting with the lowest energy configurations the flooding algorithm `barriers` explicitly builds up the basins $B(s)$ of the local minima s in the following way. For each configuration x that is read in, all its neighbors are generated and compared with a hash table that contains all previously read configurations:

(a) if x has no neighbor that was read in previously, then it is a local minimum;

(b) if x has neighbors that all belong to the basin $B(s)$ of a single local minimum s, then x also belong to $B(s)$;

(c) if x has neighbors in $k > 1$ basins $B(s_1),\dots,B(s_k)$ then it *merges* these basins, i.e., it is a saddle point. In this case, the basins $B(s_1),\dots,B(s_k)$ are united and are assigned to the deepest of these local minima.

The element x is then labeled with the local minimum to which it belongs and entered into the hash table. The run time is $O(E)$ since for each configuration we need to check each neighbor, i.e., we consider each edge of $\mathbb{G}$ twice. The effort for one step is constant due the the use of the hash table. The memory requirements are determined by the size of hash table, i.e. $O(|X|)$.

It is not hard to see that this procedure extends to partially ordered sets. The crucial observation is that for each configuration x we only need to check those neighbors y that have comparable fitness values, i.e., for which we have either $f(x) \prec f(y)$, $f(x) = f(y)$, or $f(x) \succ f(y)$.

Every partial order admits a linear extension, that is, a well-order $\dot{<}$ such that $x \prec y$ implies $x \dot{<} y$. Let us assume that X is sorted w.r.t. $\dot{<}$. When x is read then the hash contains already all configurations that strictly dominate x. On the other hand, x does not strictly dominate any of the current hash entries. Hence it suffices to check for each neighbor y of x if (i) it is contained in the hash and (ii) if it dominates x at least weakly. The configuration x belongs to the basin $B(s)$ if and only if these two conditions are satisfied. If x belongs

Algorithm 1 POSET FLOODING

Input: $\mathbb{G} = (X, E)$, $\preceq$, linear extension $\dot{<}$
Output: A barrier forest $\mathfrak{F}$ for $(X, \preceq)$

Sort X w.r.t. $\dot{<}$ in ascending order
$\mathcal{H} \leftarrow \emptyset$; /* Hash table containing all read points */
$\mathcal{B} \leftarrow \emptyset$; /* List of all "active" basins */
$\mathfrak{F} \leftarrow \emptyset$. /* Barrier Tree */
for each $x \in X$ **do**
  $Z \leftarrow \{z \in \mathbb{G}\text{-neighbors}(x) | z \in \mathcal{H} \text{ and } z \preceq x\}$.
  $B = \{b_1, \ldots, b_l\} \leftarrow \{b \in \mathcal{B} | Z \cap b \neq \emptyset\}$.
  if $|B| = 0$ **then**
    /* x is a local Pareto point */
    add x as new component to the forest $\mathcal{F}$;
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{x\}$.
  if $|B| = 1$ **then**
    /* x belongs to basin b */
    $b \leftarrow b \cup \{x\}$.
  if $|B| \geq 2$ **then**
    /* x is a "saddle" */
    record x as node connecting $b_1 \ldots b_l$ in $\mathfrak{F}$;
    $b_1 \leftarrow b_1 \cup b_2 \cup \ldots \cup b_l \cup \{x\}$;
    $\mathcal{B} \leftarrow \mathcal{B} \setminus \{b_2 \cup \ldots \cup b_l\}$.
  mark x with (a pointer to) the basin to which it belongs and enter x into the hash table $\mathcal{H}$.
determine global Pareto minima in $\mathfrak{F}$.

to more than one basin we have a saddle point in the sense defined in the previous section (it is not hard to explicitly backtrack the path **p** that connects the corresponding minima). Since basins are merged when the first saddle point that connects them is encountered, the representative of $S(x, y)$ that is chosen by this procedure depends explicitly on the linear extension $\dot{<}$. A more formal description of the poset flooding algorithm is given as Algorithm 1.

It is important to notice that the flooding algorithm always produces a tree in the linearly ordered case. For partially ordered sets, however, we only obtain a forest. In particular, if $\mathbb{Y}$ is an anti-chain, the barrier forest is the completely disconnected graph.

The barrier forest can be computed efficiently provided (i) we know a linear extension of the partial order $\preceq$ and (ii) we can efficiently decide whether or not $x \prec y$ for all pairs x and y. The run-time of the poset version of the algorithm is therefore $O(|E|q)$ where q is the number of operations necessary for checking domination. Since the computation of a linear extension from an arbitrary poset requires in the worst case $O(|X|^2)$ steps this part will in general dominate the CPU requirements.

It is interesting to compare the performance of Algorithm 1 with the approach of Kung, Luccio & Preparata [15] for finding the global Pareto points in a set of K-dimensional vectors. The latter task can be performed in $O(|X|\log^{\min(1,K-1)}|X|$ steps.

For poset landscapes arising form multi-objective minimization problems with K objective functions, however, the auxiliary function

$$\tilde{f}(x) = \sum_{i=1}^{K} a_i f_i(x) \tag{5}$$

with arbitrary constants $a_i > 0$ is a linear extension of $\prec$. Furthermore $q = O(K)$ since we have to make one comparison with each of the functions f_i. Hence the linear extension can be computed in $O(K|X|)$ time, sorting requires $O(|X|\ln|X|)$. The classes of graphs that are of interest as search spaces satisfy $|E| = O(|X|\ln^p|X|)$ with some $p \geq 1$, in most cases $p = 1$. The total runtime of Algorithm 1 is then $O(K|E|) = O(K|X|\log^p|X|)$ and hence for $p = 1$ coincides with the theoretical lower bound [15].

4. Global Pareto Points

As a by-product of the flooding algorithm we can determine the set $\mathcal{G}$ of global Pareto points as well. It is more efficient, however, to extract $\mathcal{G}$ from the barrier forest. The following, rather trivial observation, is the basis for our approach.

LEMMA 1. *A vertex x is either a Pareto minimum or it is strictly dominated by a local Pareto minimum.*

Proof. Suppose x is not a Pareto minimum. Then there is $y \in X$ such that $y \prec x$. If y is not a local Pareto minimum then it has a neighbor y' that satisfies $y' \prec y$ and hence also $y' \prec x$. If y' is not locally Pareto minimal we can repeat the argument to find a neighbor y'' of y' and so on. However, since $x \not\prec x$ and X is finite, this sequence must terminate after a finite number k of steps with a local Pareto minimum $y^{(k)} \prec x$.

As a consequence, we can identify the Pareto minima from the set $\mathcal{M}$ of local Pareto minima, i.e., from the tips of the barrier forest alone. The computational effort is $O(|\mathcal{M}|^2)$.

5. Computational Examples

An overview of the existing literature on multi-objective optimization problems can be found in [7]. The most important classes of problems include shortest path problems, assignment problems and their generalization, and knapsack problems.

5.1. Knapsack Problems

In the simplest version a *knapsack problem* [16, 17] consists of a set of n objects with values $v_i > 0$ and weights $w_i > 0$. The task is to chose a subset $J \subseteq \{1,\dots,n\}$ that simultaneously minimizes

$$f_1(J) = \sum_{i\in J} v_i \qquad \text{and} \qquad f_2(J) = -\sum_{i\in J} w_i \tag{6}$$

It is natural to consider the knapsack problem on the hypercube, i.e., each configuration is a string x of length n where $x_i = 1$ if $i \in J$ and $x_i = 0$ if $i \notin J$. The neighborhood is defined by flipping a bit in the string x, i.e., by adding or removing an item to or from the knapsack. If $q \notin J$ then $f_1(J \cup \{q\}) = f_1(J) + v_q > f_1(J)$ and $f_2(J \cup \{q\}) = f_2(J) - w_q < f_2(J)$, while for $q \in J$ we have $f_1(J \setminus \{q\}) = f_1(J) - v_q < f_1(J)$ and $f_2(J \setminus \{q\}) = f_2(J) + w_q > f_2(J)$. Thus neighboring conformations on the boolean hypercube are *always* incomparable, i.e., every configuration is a local Pareto point. The the barrier forest is therefore trivial, see Fig. 1a.

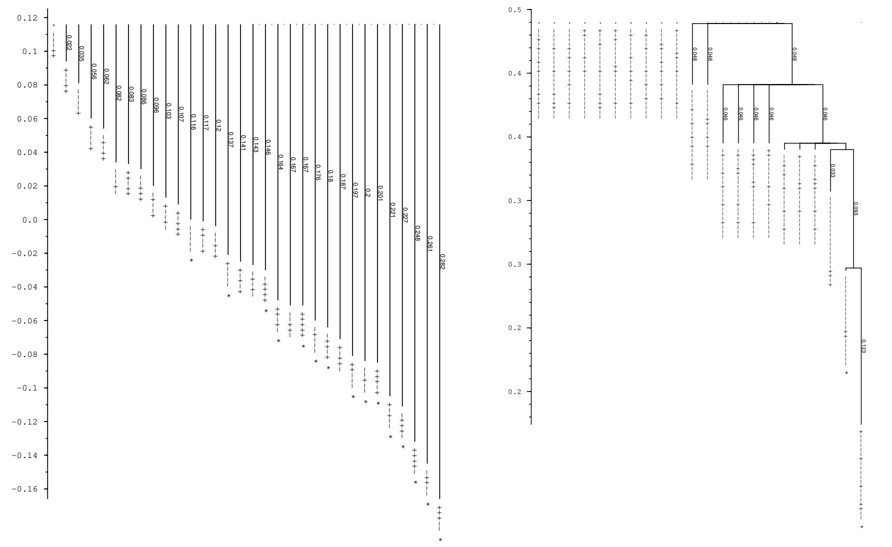
5.2. A Web Access Problem

The task of a *Web Access Problem* is to retrieve a list $\mathfrak{L}$ of documents from a set S of servers [8]. Each server $s \in S$ hosts a collection $\mathfrak{D}_s \subseteq \mathfrak{L}$ of these documents. Furthermore, there is time delay t_s and cost c_s associated with accessing s. The task is retrieve all documents of $\mathfrak{L}$ such that the total cost and the maximal waiting time of a query $Q \subseteq S$, represented by set Q of hosts that are accessed, is minimized. Again we may encode Q as a binary string x, $x_s = 1$ if $s \in Q$ and $x_s = 0$ otherwise. Flipping a bit, i.e., adding or removing a server from a query appears to be the natural definition of the neighborhood, which is used in the computational example.

The quality of an answer is defined by the union

$$\mathfrak{R}(Q) = \bigcup_{s\in Q} \mathfrak{D}_s \tag{7}$$

of documents that are retrieved. One may use $|\mathfrak{R}(Q)|$ as a quality measure in which case we again obtain a knapsack type problem. Instead we require



(a) (b)

Figure 1. Barrier forests of two small examples. The global Pareto points are marked by asterisks. The abscissa gives $\tilde{f}(x) = f_1(x) + f_2(x)$.
(a) The barrier forest for the simple 0/1 knapsack problem (5 items with randomly assigned weights and values) is trivial because every configuration (+ means $i \in J$, – means $i \notin J$) is a local Pareto point.
(b) Barrier forest of a Web Access Problem with $n = 20$ servers, $|\mathcal{L}| = 100$ documents, and $q_{\max} = 0.1$. Here the two global Pareto points are located in a common subtree.

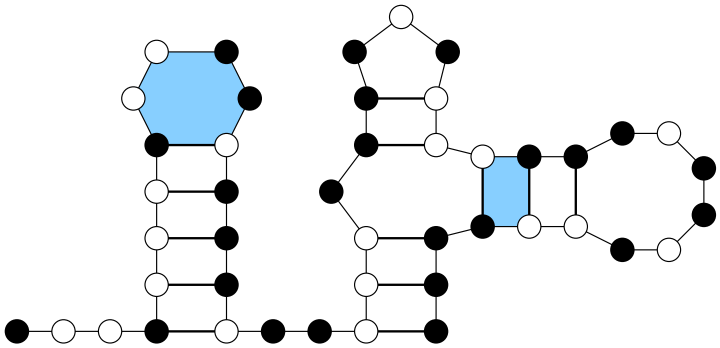
that all documents in $\mathfrak{L}$ must be retrieved and consider all other queries as failures. The two cost functions are

$$C(Q) = \sum_{s \in Q} c_s \qquad T(Q) = \max_{s \in Q} t_s \tag{8}$$

We model the individual hosts by randomly assigning a document $d \in \mathfrak{D}_s$ with a site dependent probability q_s. The values of q_s are themselves random numbers uniformly distributed in $[0, q_{\max}]$. An example is shown as Fig. 1b. Note the difference of the tree structure in comparison with the knapsack problem.

5.3. RNA Secondary Structures

An RNA molecule is a linear polymer composed of so-called nucleotides that can be represented by a string from the alphabet $\{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{U}\}$. RNAs form complex three-dimensional structures that is dominated by specific base pairs forming the molecule's *secondary structure*. Consider an RNA sequence



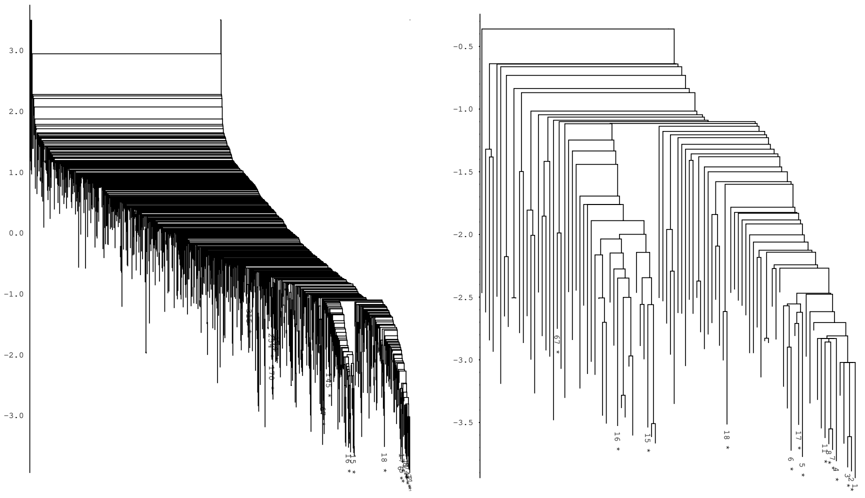
```
...(((((....)))))..(((.((...))(((......))))))
```

Figure 2. Secondary structures are outer-planar graphs, i.e., they can be drawn in the plane such that all base pair are below the outline and there are no crossings. The energy E of a sequence x and particular structure is given as the sum of contributions from the “loops” (planar faces), two of which are shaded here. Stabilizing contributions arise from so-called stacks or parallel base pairs, which correspond to quadrangles in the graph, while all other loops lead to destabilizing energy contributions.
The dot-parenthesis notation (below), which represents unpaired positions as dots and base pairs as matched pair of parentheses, is used as a convenient string representation of secondary structures by the `Vienna RNA Package`.

$x = (x_1, x_2, \ldots, x_n)$ of length n. We write (i, j) for a base pair between the sequence positions i and j, where (x_i, x_j) is one of the six allowed pairs $\{\mathsf{AU}, \mathsf{UA}, \mathsf{CG}, \mathsf{GC}, \mathsf{GU}, \mathsf{UG}\}$. The base pairs satisfy the following three constraints:
(i) Each nucleotide takes part in at most one base pair.
(ii) If (i, j) is base pair, then $|i-j| > 3$.
(iii) If (i, j) and (k, l) are base pairs and $i < k < j$ then $i < l < j$.
An example of an RNA secondary structure is shown in Fig. 2.

The conformational energy E of an RNA molecule depends both on its sequence and its structure. The standard energy model assumes that E is the sum of contribution of the “loops”, i.e., the faces of the planar representation in Fig. 2. These contributions have been obtained from detailed measurements, see e.g. [18]. The secondary structure that minimizes E can be computed efficiently using a dynamic programming algorithm [24].

Not only the thermodynamics but also the kinetics of RNA folding can be studied at the level of secondary structures. In order to investigate the dynamics of the folding process one considers the insertion or removal of a base pair as elementary step of the dynamics [9]. The transition probabilities for such steps are also be derived from measured energy parameters. For a recent review of the various aspects of the RNA world *in silico* see [10].



(a) (b)

	Sequence	Structure	ΔG	$\ln\tau$	$\tilde{f}$	B	min
1	CGGGCCGCGGCCCG	(((((....)))))	-9.7	1.279	-3.944		
2	GGGGCCGCGGCCCC	(((((....)))))	-10.6	2.084	-3.890	0.869	1
3	GGGGCCCGGGCCCC	(((((....)))))	-10.6	2.136	-3.858	0.245	2
4	CGGCCCGCGGGCCG	(((((....)))))	-9.7	1.501	-3.809	1.000	1
5	CGCCCCCCGGGGCG	(((((....)))))	-9.7	1.556	-3.775	1.306	1
6	GGCCCCGCGGGGCC	(((((....)))))	-10.6	2.354	-3.725	1.171	5
7	GGGCCCGCGGGCCC	(((((....)))))	-10.6	2.364	-3.719	0.889	4
8	GGGCCCCGGGGCCC	(((((....)))))	-10.6	2.438	-3.673	0.122	7
11	GGGCCGCGCGGCCC	(((((....)))))	-9.7	1.802	-3.624	0.766	7
15	GCCCGGCGCCGGGC	(((((....)))))	-9.7	1.940	-3.540	0.980	**10**
16	GCCGGGCGCCCGGC	(((((....)))))	-9.7	1.959	-3.528	1.180	**13**
17	CGCCCGGCCGGGCG	(((((....)))))	-8.8	1.257	-3.519	0.898	5
18	CGCCGGGCCCGGCG	(((((....)))))	-8.8	1.262	-3.516	1.916	1
67	GCCCCCGCGGGGGC	(((((....)))))	-10.6	3.941	-2.753	0.765	**37**
145	CCCGGGCGCGCCCG	..((((....))))	-6.1	1.256	-2.204	0.887	**112**
170	CCCGGCGCGCGCCG	..((((....))))	-5.8	1.217	-2.082	1.184	**69**
244	GGGCCGGGCGGCGC	..((((....))))	-4.8	1.215	-1.595	0.845	1
254	CCCGGCCCGCGCCG	..((((....))))	-4.7	1.204	-1.553	0.488	170
315	CGCCGGGCCCCCGC	...((((....))).	-3.7	1.071	-1.147	0.568	1

Figure 3. RNA secondary structure folding landscape of the GC sequences of length $n = 14$; see Fig. 2 for the explanation of the dot-parenthesis notation for RNA secondary structures. Objective functions are the thermodynamic stability of the ground state structure ΔG and the (logarithm of) the expected folding time $\ln\tau$. Point mutations $\texttt{G} \leftrightarrow \texttt{C}$ define the move set.
(a) Full tree. (b) Detail of the 100 local Pareto points with the lowest values of $\tilde{f}$. Global Pareto points are labeled. Detailed information on the global Pareto points is compiled below. B denotes the height of the barrier w.r.t. $\tilde{f}$; the "min" column gives a (local) Pareto point with smaller $\tilde{f}$ than is reachable across the barrier; structures are displayed in dot-parenthesis notation, see Fig. 2. non-global Pareto points are shown in bold.

One interesting question about biopolymer folding is the relationship between thermodynamic stability and foldability. The latter can be measured for instance by the expected time that is need for the open chain (which has no base pairs) to reach the ground state structure. In particular, one is interested in those sequences that are both exceptionally stable and fold rapidly. It seems natural, therefore to consider the poset-valued landscape in which $f_1(x) =$

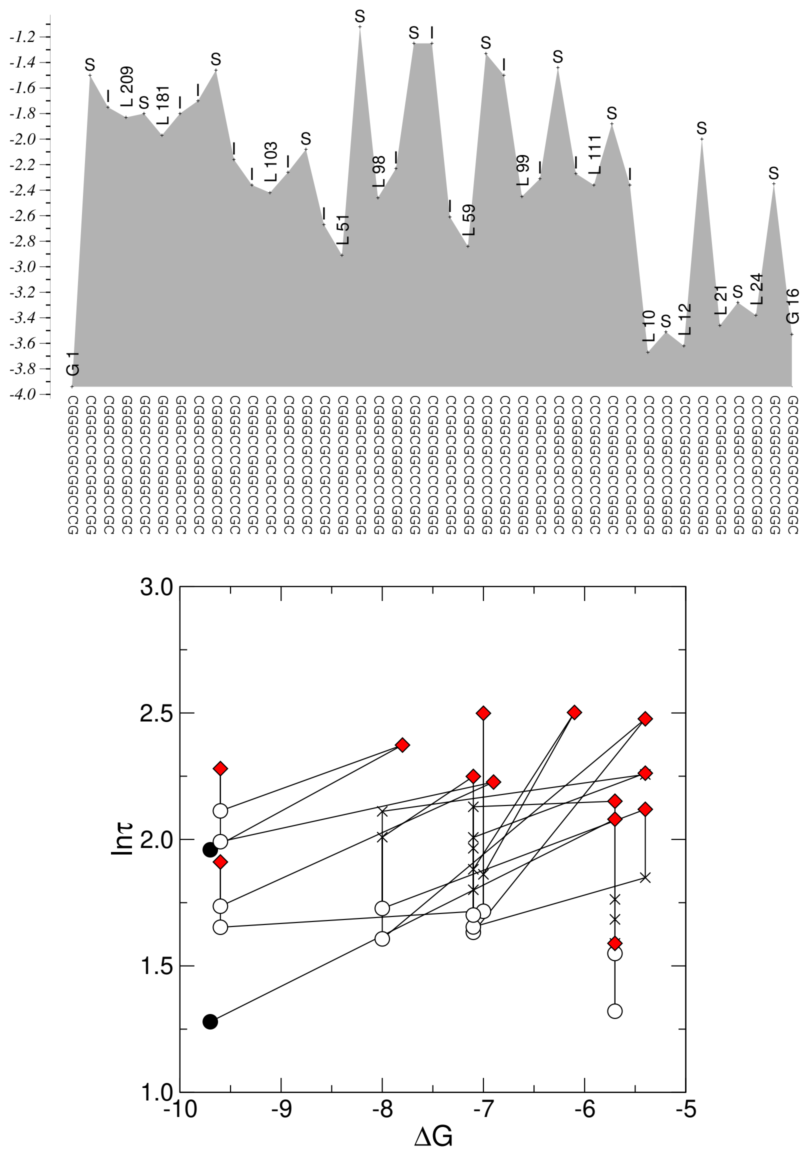


Figure 4. The path connecting the global Pareto points 1 and 16 in the GC landscape of Fig. 3 visualizes the rugged structure of the landscape. Above: Local or global Pareto points are labeled by L or G, resp., and their rank w.r.t. $\tilde{f}$, saddle points are indicated by S and appear as local maxima along the connecting path. Intermediate sequences along connections between saddles and Pareto points are indicated by I.
Below: The same path in the $\Delta G/\ln\tau$ plane. Local Pareto points are shown as circles (full for the two global Pareto points), ⧫ marks saddle points, and intermediate points are shown as crosses.

$\Delta G(x)$ is the ground state energy of the sequence x and $f_2(x) = \ln \tau(x)$ is the (logarithm of) the expected folding time. For the computations reported here the programs `RNAfold` (to obtain the ground state structure) and `kinfold` (for simulating folding trajectories) were used. Both are part of the `Vienna RNA Package` and can be obtained free of charge on the internet.[2]. The folding times reported here are averages over 1000 runs of `kinfold` for each sequence. The auxiliary function is defined as $\tilde{f}(x) = \Delta G(x)/s' + \ln \tau(x)/s''$ where s' and s'' are the standard deviations of the $\Delta G(x)$ and $\ln \tau(x)$ across all sequences. In order to keep the computational efforts at a reasonable level we restrict ourselves to GC-only sequences. It is argued e.g. in [13] that the RNA landscapes with reduced alphabets share the qualitative properties with the full biophysical GCAU-alphabet.

In Fig. 3 we summarize the results for GC sequences of length 14. Among the 16384 sequence we find 720 local Pareto points of which 19 are global Pareto points. They fall roughly into two classes: those that are close to the global optimum of thermodynamic stability share the most stable secondary structures, and those that fold exceptionally fast also sharing a common structural motif. The rugged structure of the landscape is reflected by the geometry of the paths that connect Pareto points with each other, Fig. 4.

For larger landscapes, the number of Pareto points and connecting saddles increases exponentially so that a graphical representation soon becomes infeasible. Nevertheless, we see that the global Pareto points also fall into a small number of classes that are characterized by their secondary structures. It is interesting to note that the global Pareto points are widely separated in sequences space; connecting paths typically run through a large number of saddles and local Pareto points, as shown in Fig. 4.

6. Discussion

The (local) Pareto points of a multi-objective optimization problem can be seen as the generalization of the (local) optima of a fitness landscape. In this contribution we have shown that the notion of a saddle point and the concept of paths connecting local optima can also be transferred from conventional, real valued, landscapes to poset-valued landscapes. This implies that barrier trees can be defined for multi-objective optimization problems. We have described an algorithm that generates these trees and an efficient implementation that allows us to study the landscapes of moderate size problems in great detail. As the focus of this contribution is the method rather than its application we provide only a few examples to illustrate the type of information that can be gained.

[2] `http://www.tbi.univie.ac.at/RNA`

In contrast to the well-ordered case the “barrier trees” need not be connected in the case of posets. Disconnectedness points at a particularly rugged structure and indicates that the move set is not suitable for local search. For instance, packing or unpacking single items in a knapsack problem will always produce incomparable weight/value pairs. Other move-sets might be much better suited for this class of problems. The structure of the barrier trees for small instances could be used to compare difference move sets. Move sets that reduce the number of local Pareto points and/or the height of the saddles points that separate them can be exprected to lead to improved heuristics also for larger instances of the same problem class.

Another application is the systematic investigation of the relationships between different properties of biopolymers. As an example we have considered here the thermodynamic and kinetic properties of RNA folding. We find that global Pareto points are widely separated in sequence space, i.e., there are no indications for a substantial clustering of stable, fast-folding sequences.

The current approach produces a single representative of the saddle point set $S(x,y)$ that depends explicitly on the choice of the linear extension $\dot{<}$ of the poset. It would be of interest to list the complete set $S(x,y)$ instead. It is not clear, however, whether the flooding algorithm can be modified to do this efficiently.

The definition of saddle points by means of paths that are constrained such that its points are locally comparable can in principle be extended to continuous search spaces; we suspect, however, that it will not be possible to compute them efficiently from the path-based definition.

The approach presented here is intended as a tool for analyzing a landscape that is known exhaustively. In principle, it could be turned into a method for optimization. In the linearly ordered case the idea is the following: As part of the flooding algorithm we explicitly construct all neighbors of a configuration. At this point we might in addition evaluate their cost function. If a neighbor y has a cost that is smaller than the current point, we store it in a “waiting list”, otherwise we insert it into the list of unread points at the appropriate position. We stop the flooding algorithm at a maximum value of the cost function (or when all basins are connected). At this point the “waiting list” contains entry points to basins that were not represented in the beginning. We use gradient descent to find the local minima associated with “waiting list” configurations and repeat the flooding algorithm. The entire procedure is iterated until the “waiting list” produced by the modified flooding algorithm is empty. The same approach could be applied to the multi-objective case using Algorithm 1 as the starting point.

Availablity

Algorithm 1 is implemented in the current version 1.0.0 of the program `barriers` which is available from `http://www.tbi.univie.ac.at/TBI/software.html`.

Acknowledgments

The idea to this research originated during the Dagstuhl meeting on Evolutionary Computation in January 2002. We are grateful to all participants for stimulating discussions.

References

1. Azencott, R.: 1992, *Simulated Annealing*. New York: John Wiley & Sons.
2. Catoni, O.: 1999, 'Simulated Annealing Algorithms and Markov Chains with Rate Transitions'. In: J. Azema, M. Emery, M. Ledoux, and M. Yor (eds.): *Seminaire de Probabilites XXXIII*, Vol. 709 of *Lecture Notes in Mathematics*. Berlin/Heidelberg: Springer, pp. 69–119.
3. Coello, C. A.: 2000, 'An updated survey of GA-based multiobjective optimization techniques'. *ACM Comput. Surveys* **32**, 109–143.
4. Dasgupta, P., P. P. Chakrabarti, and S. C. DeSakar: 1999, *Multiobjective Heuristic Search*. Braunschweig, Germany: Vieweg.
5. Davey, B. A. and H. A. Priestley: 1990, *Introduction to Lattice and Order*. Cambridge UK: Cambridge Univ. Press.
6. Deb, K.: 2001, *Multi-Objective Optimization using Evolutionary Algorithms*. Chichester, NY: Wiley.
7. Ehrgott, M. and X. Gandibleux: 2000, 'A survey and annotated bibliography of multiobjective combinatorial optimization'. *OR Spektrum* **22**, 425–460.
8. Etzioni, O., S. Hanks, T. Jiang, R. M. Karp, O. Madari, and O. Waarts: 1996, 'Efficient Information Gathering on the Internet'. In: *37th Annual Symposium on Foundations of Computer Science*. pp. 234–243. Burlington, Vermont, 14-16 October 1996.
9. Flamm, C., W. Fontana, I. Hofacker, and P. Schuster: 2000, 'RNA folding kinetics at elementary step resolution'. *RNA* **6**, 325–338.
10. Flamm, C., I. L. Hofacker, and P. F. Stadler: 1999, 'RNA *in silico*: The Computational Biology of RNA Secondary Structures'. *Adv. Complex Syst.* **2**, 65–90.
11. Flamm, C., I. L. Hofacker, P. F. Stadler, and M. T. Wolfinger: 2002, 'Barrier Trees of Degenerate Landscapes'. *Z. Phys. Chem.* **216**, 155–173.
12. Fonseca, C. M. and P. J. Fleming: 1993, 'Genetic Algorithms for multi-objective optimization: formulation, discussion, and generalization'. In: S. Forrest (ed.): *Proceedings of the Fifth International Conference on Genetic Algorithms*. San Francisco CA, pp. 416–423.
13. Grüner, W., R. Giegerich, D. Strothmann, C. M. Reidys, J. Weber, I. L. Hofacker, P. F. Stadler, and P. Schuster: 1996, 'Analysis of RNA Sequence Structure Maps by Exhaustive Enumeration. I. Neutral Networks'. *Monath. Chem.* **127**, 355–374.
14. Iacob, P.: 1986, 'Saddle point duality theorem for Pareto optimization'. *Anal. Num. Theorie Approx.* **15**, 37–40.

15. Kung, H. T., F. Luccio, and F. P. Preparata: 1975, 'On Finding the Maxima of a Set of Vectors'. *J. Assoc. Comp. Mach.* **22**, 469–476.
16. Lagoudakis, M. G.: 1996, 'The 0-1 Knapsack Problem — An Introductory Survey'. `citeseer.nj.nec.com/151553.html`.
17. Martello, S. and P. Toth: 1990, *Knapsack problems: Algorithms and computer implementations*. Chichester, England: Wiley.
18. Mathews, D., J. Sabina, M. Zucker, and H. Turner: 1999, 'Expanded Sequence Dependence of Thermodynamic Parameters Provides Robust Prediction of RNA Secondary Structure'. *J. Mol. Biol.* **288**, 911–940.
19. Nemoto, K.: 1988, 'Metastable states of the SK spin glass model'. *J. Phys. A* **21**, L287–L294.
20. Rammal, R., G. Toulouse, and M. A. Virasoro: 1986, 'Ultrametricity for physicists'. *Rev. Mod. Phys.* **58**, 765–788.
21. Reidys, C. M. and P. F. Stadler: 2002, 'Combinatorial Landscapes'. *SIAM Review* **44**, 3–54.
22. Van Veldhuizen, D. A. and G. B. Lamont: 2000, 'Multiobjective evolutionary algorithms: analyzing the state-of-the-art'. *Evol. Comp.* **8**, 125–147.
23. Wales, D. J., M. A. Miller, and T. R. Walsh: 1998, 'Archetypal energy landscapes'. *Nature* **394**, 758–760.
24. Zuker, M. and D. Sankoff: 1984, 'RNA secondary structures and their prediction'. *Bull. Math. Biol.* **46**, 591–621.

Address for Offprints:
P F Stadler
Bioinformatik, Institut für Informatik
Universität Leipzig
Kreuzstraße 7b
D-04103 Leipzig, Germany
Phone: +49 341 14951 20, Fax: +43 341 14951 19