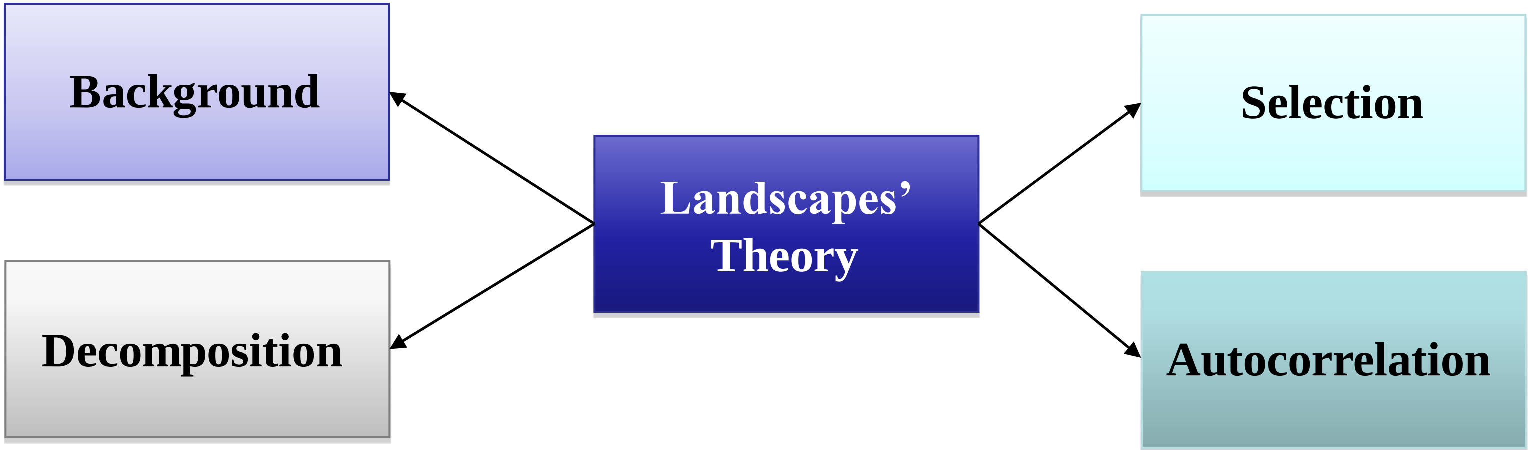
Elementary Landscape Decomposition of Combinatorial Optimization Problems

Francisco Chicano

Joint work with L. Darrell Whitley and Enrique Alba

Motivation

- **Landscapes' theory is a tool for analyzing optimization problems**
- **Applications in Chemistry, Physics, Biology and Combinatorial Optimization**
- **Central idea: study the search space to obtain information**
  - **Better understanding of the problem**
  - **Predict algorithmic performance**
  - **Improve search algorithms**



Landscape Definition

- A **landscape** is a triple (X,N,f) where
  - X is the solution space
  - N is the neighbourhood operator
  - f is the objective function

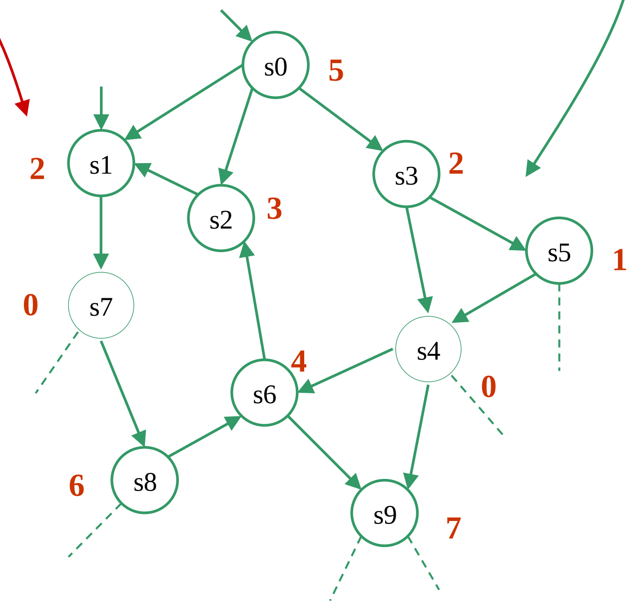
The pair (X,N) is called **configuration space**

- The **neighbourhood operator** is a function

  $N: X \rightarrow \mathcal{P}(X)$

- Solution y is **neighbour of** x if $y \in N(x)$
- **Regular** and **symmetric** neighbourhoods
  - $d=|N(x)| \quad \forall x \in X$
  - $y \in N(x) \Leftrightarrow x \in N(y)$
- **Objective function**

  $f: X \rightarrow R$ (or N, Z, Q)



Elementary Landscapes: Formal Definition

- An elementary function is an eigenvector of the graph Laplacian (plus constant)

Adjacency matrix

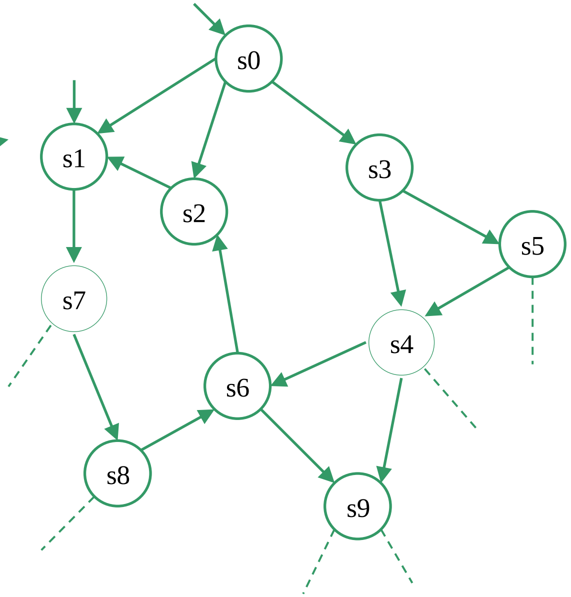
$$A_{xy} = \begin{cases} 1 & \text{if } y \in N(x) \\ 0 & \text{otherwise} \end{cases}$$

Degree matrix

$$D_{xy} = \begin{cases} |N(x)| & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$$

- Graph Laplacian: $\Delta = A - D$

Depends on the configuration space



- Elementary function: eigenvector of Δ (plus constant)

$$\Delta \times (\vec{f} - b) = \lambda \cdot (\vec{f} - b)$$

Eigenvalue

Elementary Landscapes: Characterizations

- An **elementary landscape** is a landscape for which

$$\underset{y \in N(x)}{\text{avg}}\{f(y)\} = \alpha f(x) + \beta \quad \forall x \in X$$

where

$$\underset{y \in N(x)}{\text{avg}}\{f(y)\} \stackrel{\text{def}}{=} \frac{1}{d} \sum_{y \in N(x)} f(y)$$

Depend on the problem/instance

Linear relationship

- **Grover's wave equation**

$$\bar{f} \stackrel{\text{def}}{=} \frac{1}{|X|} \sum_{y \in X} f(y)$$

$$\underset{y \in N(x)}{\text{avg}}\{f(y)\} = f(x) + \frac{k}{d}\left(\bar{f} - f(x)\right) \quad \forall x \in X$$

$$\alpha = 1 - \frac{k}{d} \qquad \beta = \frac{k}{d}\bar{f}$$

Characteristic constant: $k = -\lambda$

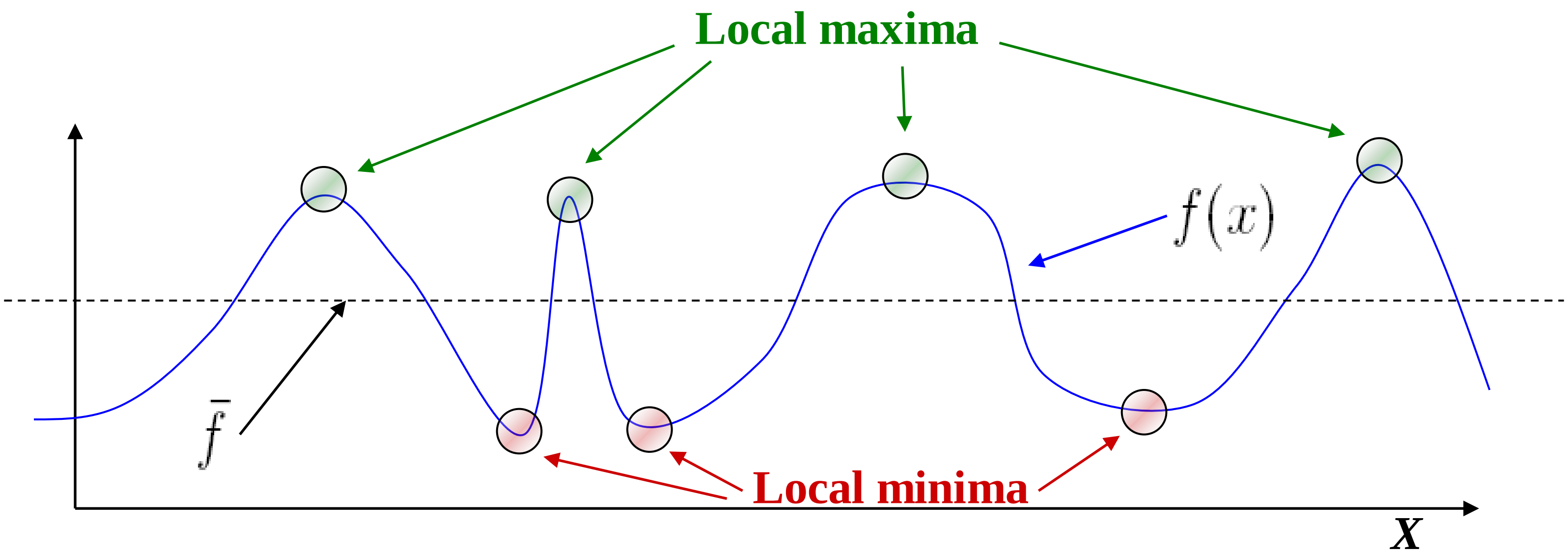
Elementary Landscapes: Properties

- **Several properties of elementary landscapes are the following**

$$f(x) < \min\left\{\underset{y\in N(x)}{\text{avg}}\{f(y)\}, \bar{f}\right\} \quad \text{or} \quad f(x) > \max\left\{\underset{y\in N(x)}{\text{avg}}\{f(y)\}, \bar{f}\right\}$$

where $f(x) \neq \bar{f}$

- **Local maxima and minima**

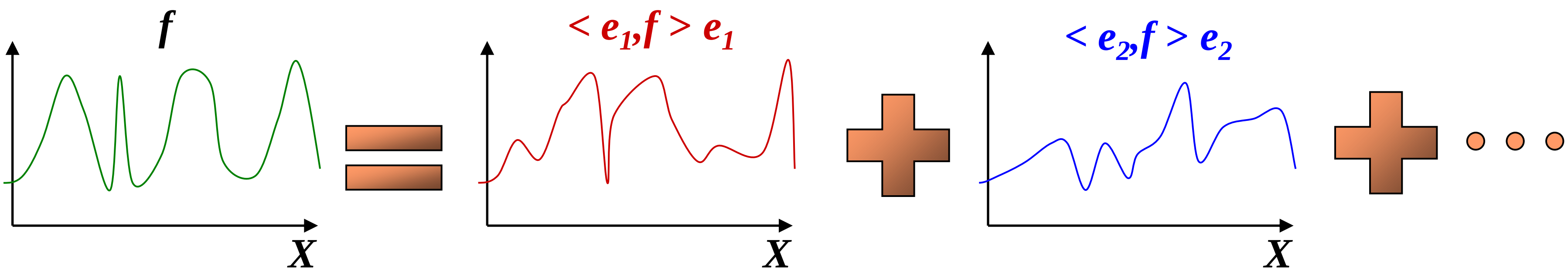


Elementary Landscapes: Examples

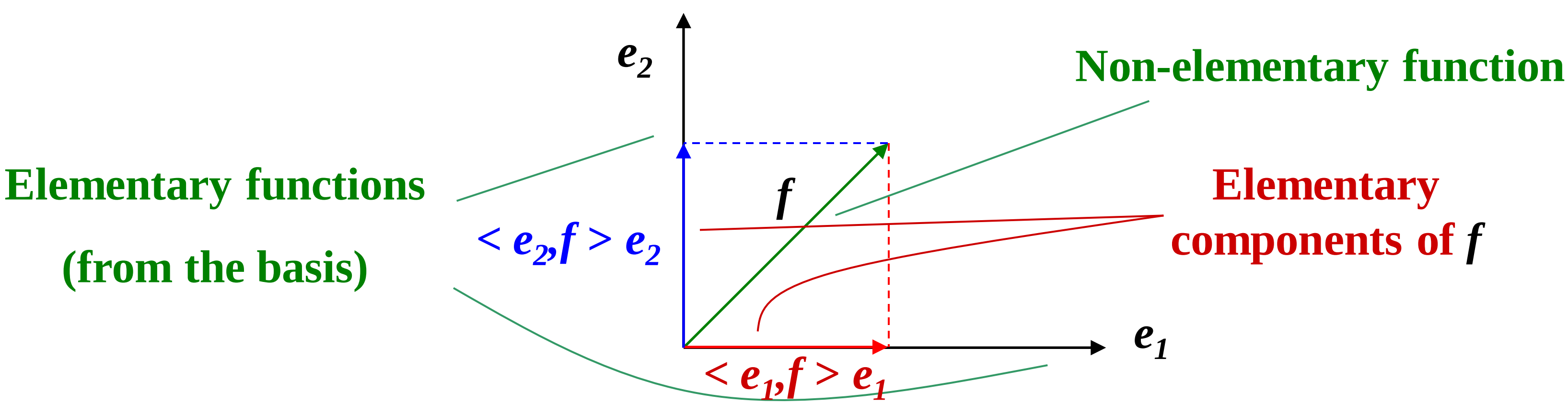
Problem	Neighbourhood	d	k
Symmetric TSP	2-opt	$n(n$-$3)/2$	n-1
	swap two cities	$n(n$-$1)/2$	$2(n$-$1)$
Antisymmetric TSP	inversions	$n(n$-$1)/2$	$n(n+1)/2$
	swap two cities	$n(n$-$1)/2$	$2n$
Graph α-Coloring	recolor 1 vertex	$(\alpha$-$1)n$	2α
Graph Matching	swap two elements	$n(n$-$1)/2$	$2(n$-$1)$
Graph Bipartitioning	Johnson graph	$n^2/4$	$2(n$-$1)$
NAES	bit-flip	n	4
Max Cut	bit-flip	n	4
Weight Partition	bit-flip	n	4

Landscape Decomposition: Overview

- **What if the landscape is not elementary?**
- **Any landscape can be written as the sum of elementary landscapes**



- **There exists a set of eigenfunctions of Δ that form a basis of the function space**



Landscape Decomposition: General Approach

- **How to decompose a function into elementary landscapes?**
- **Computing $\langle e_i, f \rangle e_i$ for a basis $\{e_1, e_2, ..., e_{|X|}\}$**
- **We need a basis**

Walsh functions (in binary strings with bit-flip neighborhood)

$$\{\psi_w\} \quad w \in \{0,1\}^n$$

$$\psi_w(x) = \frac{1}{2^n}(-1)^{\sum_{i=1}^{n} w_i x_i}$$

- **We need to compute $\langle e_i, f \rangle$ which requires a sum of $|X|$ elements in general**

$$\langle \psi_w, f \rangle = \sum_{x \in X} \psi_w(x) f(x)$$

Landscape Decomposition: Methodology (I)

- **Methodology for the decomposition that does not require a basis**

1 **Select small instances of the problem**

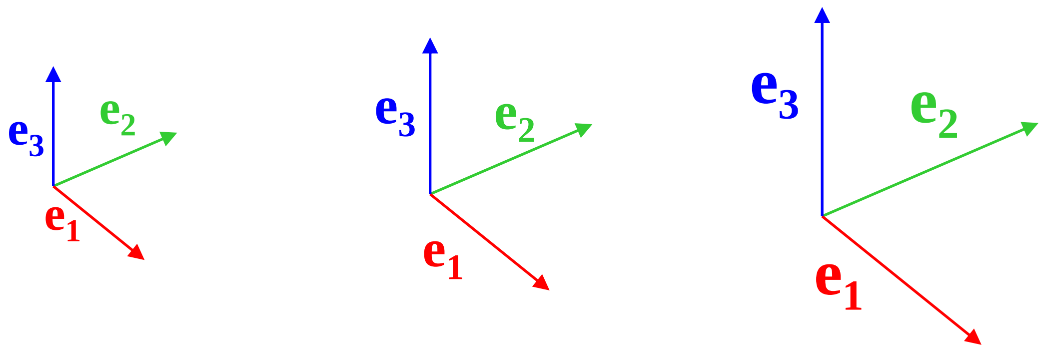



2 **Explicitly compute the Laplacian matrix and represent the objective function as a vector**

$$(\Delta_1, \vec{f_1}) \quad (\Delta_2, \vec{f_2}) \quad (\Delta_3, \vec{f_3})$$

Landscape Decomposition: Methodology (II)

3 **Find an orthonormal basis of the vector space that are eigenvectors of the Laplacian**

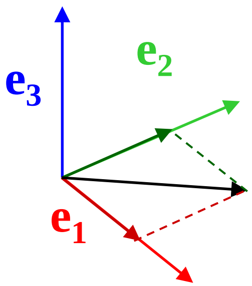


4 **Find the coordinates of the objective function in the basis**

$$\langle \vec{e}_i, \vec{f}_1 \rangle \quad \langle \vec{e}_i, \vec{f}_2 \rangle \quad \langle \vec{e}_i, \vec{f}_3 \rangle$$

Landscape Decomposition: Methodology (III)

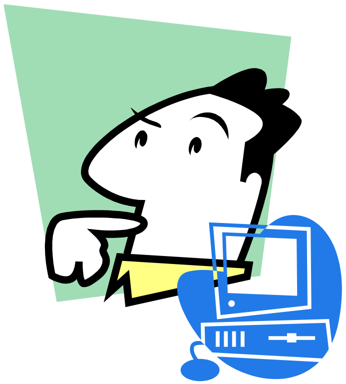
5 **Sum the components with the same eigenvalue to compute the elementary landscape decomposition**



$$\vec{f}_i = \vec{f}_i^{\lambda_1} + \vec{f}_i^{\lambda_2} + \ldots$$

6 **For each component find a generalization of the function**

$$\left.\begin{matrix} \vec{f}_1^{\lambda_1} & \vec{f}_2^{\lambda_1} & \vec{f}_3^{\lambda_1} \\ \vec{f}_1^{\lambda_2} & \vec{f}_2^{\lambda_2} & \vec{f}_3^{\lambda_2} \end{matrix}\right\} \quad f^{\lambda_1} \quad f^{\lambda_2}$$



Landscape Decomposition: Methodology (& IV)

7 **Check that the generalizaed functions are elementary**

$$\underset{y \in N(x)}{\mathrm{avg}} \{ f^{\lambda_i}(y) \} = \alpha f^{\lambda_i}(x) + \beta$$

8 **Check if the sum of the generalized functions is the objective function**

$$f \stackrel{?}{=} \sum_i f^{\lambda_i}$$

Landscape Decomposition: FAP

- Using the one-change neighborhood, the fitness function can be decomposed into two elementary components:

$$f_{2r} = \frac{1}{r} \sum_{\substack{i,j=1 \\ i \neq j}}^{n} \sum_{p,q=1}^{r} w_{i,j}^{p,q} \phi_{i,j,r-2}^{p,q}$$

$k_1 = 2r$

$$f_r = -\frac{1}{r} \sum_{\substack{i,j=1 \\ i \neq j}}^{n} \sum_{p,q=1}^{r} w_{i,j}^{p,q} \phi_{i,j,-2}^{p,q} + \sum_{i=1}^{n} \sum_{p=1}^{r} w_{i,i}^{p,p} \varphi_{i,i}^{p,p}$$

$k_2 = r$

$$f(x) = f_{2r}(x) + f_r(x)$$

Landscape Decomposition: QAP

- **Using the swap neighborhood, the fitness function can be decomposed into three elementary components:**

$$f_{c1}(x) = \sum_{\substack{i,j,p,q=1 \\ i \neq j \\ p \neq q}}^{n} r_{ij} w_{pq} \frac{\Omega^1_{(i,j),(p,q)}(x)}{2n}$$

$k_1 = 2n$

$$f_{c2}(x) = \sum_{\substack{i,j,p,q=1 \\ i \neq j \\ p \neq q}}^{n} r_{ij} w_{pq} \frac{\Omega^2_{(i,j),(p,q)}(x)}{2(n-2)}$$

$k_2 = 2(n\text{-}1)$

Kronecker's delta

$$f_{c3}(x) = \sum_{\substack{i,j,p,q=1 \\ i \neq j \\ p \neq q}}^{n} r_{ij} w_{pq} \frac{\Omega^3_{(i,j),(p,q)}(x)}{n(n-2)} + \sum_{i,p=1}^{n} r_{ii} w_{pp} \delta^p_{x(i)}$$

$k_3 = n$

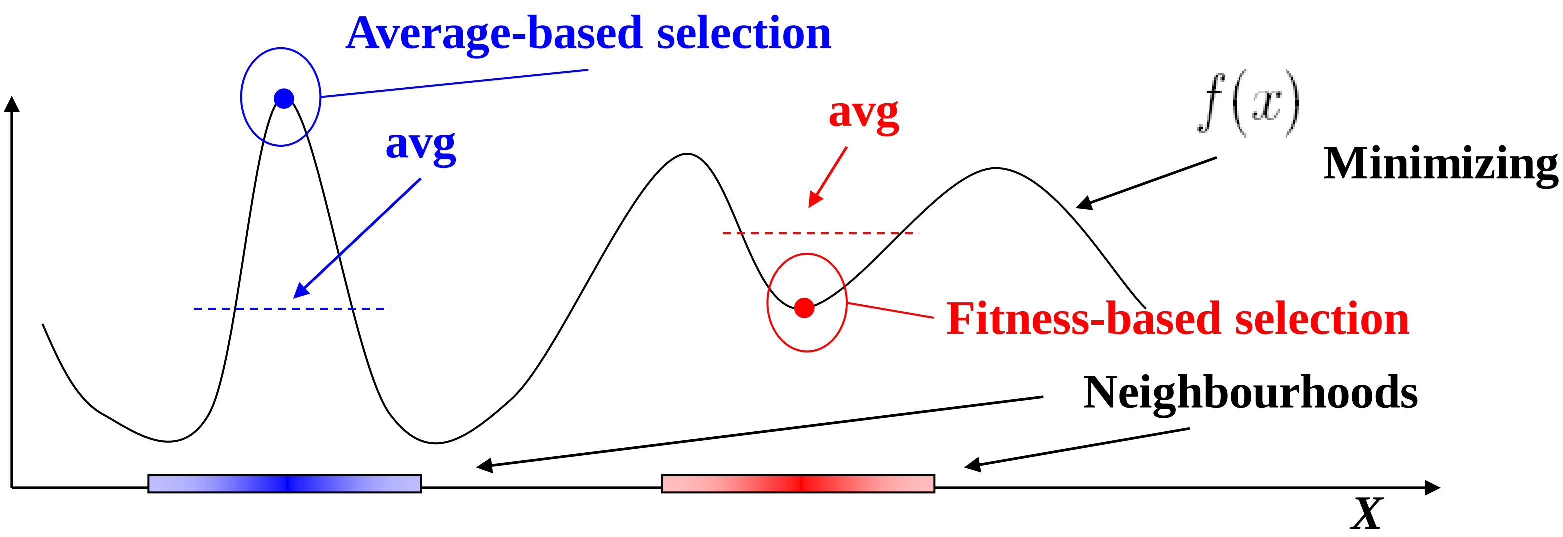
$$f(x) = f_{c1}(x) + f_{c2}(x) + f_{c3}(x)$$

Landscape Decomposition: Examples

Problem	Neighbourhood	*d*	Components
General TSP	inversions	$n(n-1)/2$	2
	swap two cities	$n(n-1)/2$	2
Subset Sum Problem	bit-flip	n	2
MAX k-SAT	bit-flip	n	k
NK-landscapes	bit-flip	n	$k+1$
Radio Network Design	bit-flip	n	max. nb. of reachable antennae
Frequency Assignment	change 1 frequency	$(\alpha-1)n$	2
QAP	swap two elements	n(n-1)/2	3

New Selection Strategy

- **Selection operators usually take into account the fitness value of the individuals**



- **We can improve the selection operator by selecting the individuals according to the average value in their neighbourhoods**

New Selection Strategy

- **In elementary landscapes the traditional and the new operator are (almost) the same!**

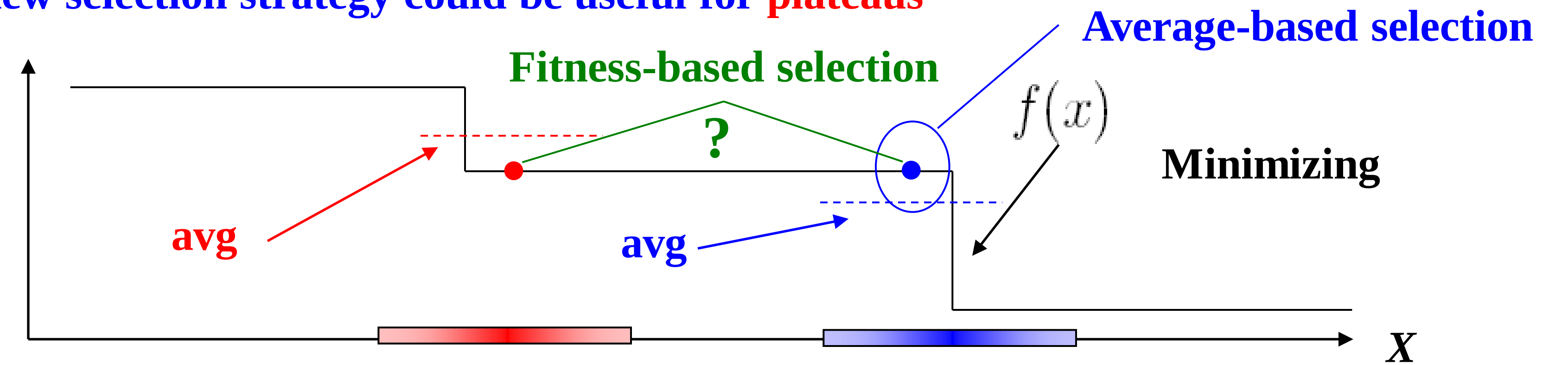
Recall that...
$$\underset{y \in N(x)}{\operatorname{avg}}\{f(y)\} = \alpha f(x) + \beta \quad \forall x \in X$$

- **However, they are not the same in non-elementary landscapes. If we have *n* elementary components, then:**

$$\underset{y \in N(x)}{\operatorname{avg}}\{f(y)\} = \alpha_0 + \alpha_1 f(x) + \sum_{i=2}^{n} \alpha_i f_i(x) \quad \forall x \in X$$

Elementary components

- **The new selection strategy could be useful for plateaus**



Autocorrelation

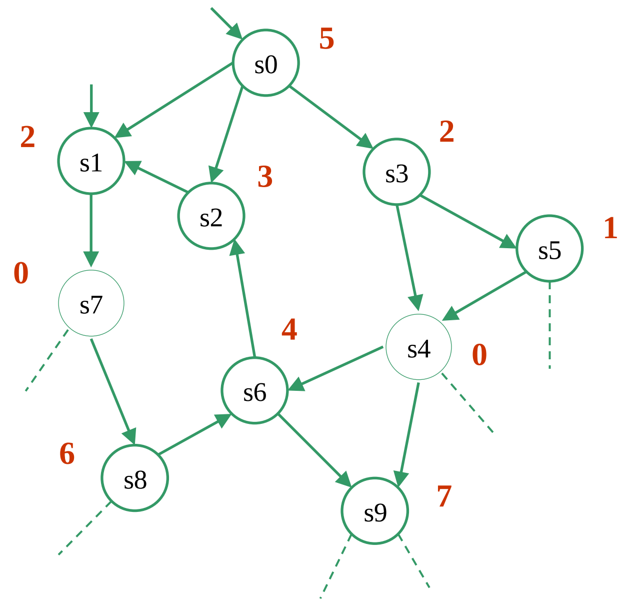
- **Let $\{x_0, x_1, ...\}$ a simple random walk on the configuration space where $x_{i+1} \in N(x_i)$**
- **The random walk induces a time series $\{f(x_0), f(x_1), ...\}$ on a landscape.**
- **The autocorrelation function is defined as:**

$$r(s) = \frac{\langle f(x_t) f(x_{t+s}) \rangle_{x_0,t} - \langle f \rangle^2}{\langle f^2 \rangle - \langle f \rangle^2}$$

- **The autocorrelation length and coefficient:**

$$l = \sum_{s=0}^{\infty} r(s) \qquad \xi = \frac{1}{1 - r(1)}$$

- **Autocorrelation length conjecture:**



The number of local optima in a search space is roughly

$$M \approx |X| / |X(x_0, l)|$$

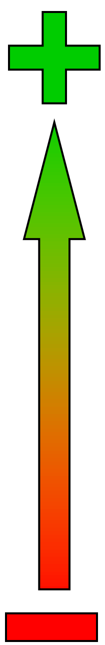
Solutions reached from x_0 after l moves

Autocorrelation Length Conjecture

- **The higher the value of *l* and ξ the smaller the number of local optima**
- ***l* and ξ is a measure of ruggedness**

Angel, Zissimopoulos. Theoretical Computer Science 263:159-172 (2001)

Length Coefficient



Ruggedness	SA (configuration 1)		SA (configuration 2)	
	% rel. error	nb. of steps	% rel. error	nb. of steps
$9.5 \leq \xi < 9.0$	0.2	50,500	0.1	101,395
$9.0 \leq \xi < 8.5$	0.3	53,300	0.2	106,890
$8.5 \leq \xi < 8.0$	0.3	58,700	0.2	118,760
$8.0 \leq \xi < 7.5$	0.5	62,700	0.3	126,395
$7.5 \leq \xi < 7.0$	0.7	66,100	0.4	133,055
$7.0 \leq \xi < 6.5$	1.0	75,300	0.6	151,870
$6.5 \leq \xi < 6.0$	1.3	76,800	1.0	155,230
$6.0 \leq \xi < 5.5$	1.9	79,700	1.4	159,840
$5.5 \leq \xi < 5.0$	2.0	82,400	1.8	165,610

Autocorrelation and Landscapes

- **If f is a sum of elementary landscapes:**

Fourier coefficients

$$r(s) = \sum_{i \neq 0} \frac{a_i^2}{\sum_{j \neq 0} a_j^2} \left(1 - \frac{k_i}{d}\right)^s \qquad l = \sum_{i \neq 0} \frac{d\ a_i^2}{k_i \sum_{j \neq 0} a_j^2}$$

- **Summing all the squared coefficients with the same k_i:**

$$r(s) = \sum_i W_i \left(1 - \frac{k_i}{d}\right)^s \qquad \xi = \frac{d}{\sum_i W_i k_i} \qquad l = d \sum_i \frac{W_i}{k_i}$$

where

$$W_i = \frac{\overline{f_{ci}^2} - \overline{f_{ci}}^2}{\overline{f^2} - \overline{f}^2}$$

Autocorrelation for FAP and QAP

- **Using the landscape decomposition we can compute l and ξ**

FAP

$$\xi = \frac{n(r-1)}{r(2-W_2)} \qquad l = \frac{n(r-1)(1+W_2)}{2r}$$

- **W_2 can be computed in polynomial time, $O(n^4 r^4)$**

QAP

$$\xi = \frac{n(n-1)}{(4-2W_3)n - 4W_2} \qquad l = \frac{(1+W_3)(n-1)+W_2}{4}$$

- **W_2 and W_3 computed in $O(n^8) \rightarrow O(n^2)$ (optimal complexity)**
- **We computed l and ξ for all the instances in the QAPLIB**

Autocorrelation for FAP and QAP

- Using the la…

$$\xi = \ldots + W_2)$$

- W_2 can be c…

$$\xi = \frac{\ldots + W_3)(n-1) + W_2}{(4 \ldots \quad 4}$$

- W_2 and W_3 c… optimal complexity)
- We computed ℓ and ξ for all the instances in the QAPLIB

Instance	ξ	ℓ	Instance	ξ	ℓ
bur26a	11.825	12.130	esc32b	8.000	8.000
bur26b	11.727	12.073	esc32c	8.000	8.000
bur26c	12.109	12.291	esc32d	8.000	8.000
bur26d	12.050	12.258	esc32e	8.000	8.000
bur26e	12.032	12.248	esc32f	8.000	8.000
bur26f	11.962	12.208	esc32g	8.000	8.000
bur26g	12.323	12.407	esc32h	8.000	8.000
bur26h	12.296	12.392	esc64a	16.000	16.00[illegible]
chr12a	3.096	3.171	had12	3.743	4[illegible]
chr12b	3.201	3.346	had14	4.319	4.[illegible]
chr12c	3.044	3.079	had16	4.405	[illegible]
chr15a	3.917	4.049	had18	5.08[illegible]	[illegible]
chr15b	4.126	4.388	had20	[illegible]	[illegible]
chr15c	3.843	3.920	kra30a	[illegible]	[illegible]
chr18a	4.585	4.658	kra30[illegible]	[illegible]	[illegible]
chr18b	4.632	4.742	[illegible]	[illegible]	[illegible]
chr20a	5.105	5.195	[illegible]	[illegible]	[illegible]
chr20b	5.035	5.067	lipa2[illegible]	[illegible]	[illegible]
chr20c	5.260	5.4[illegible]	[illegible]	[illegible]	[illegible]

Conclusions & Future Work

Conclusions

- Elementary landscape decomposition is a useful tool to understand a problem
- The decomposition can be used to design new operators and search algorithms
- We can exactly determine the autocorrelation functions
- We propose a methodology for the decomposition

Future Work

- Search for additional applications of landscapes' theory in EAs
- Design new operators and search methods based on landscapes' information
- Analyze other problems

Thanks for your attention !!!