Nature-Inspired Search and Optimisation
(and its variants(*))

Lecture 3: Representation

Christine Zarges
School of Computer Science
University of Birmingham
c.zarges@cs.bham.ac.uk

(*) 06-28209 (10 cr, 3/H), 06-27818 (20 cr, 3/H), 06-28211 (10 cr, 4/M), 06-27819 (20 cr, 4/M)
Today’s Plan

1. Representation

2. Summary
Problem Solving as a Search Task

Recall

Definition: Optimisation Problem II

Goal: For a given instance \((S, f)\) find a solution \(s^* \in S\) that is globally optimal, i.e., \(f(s^*) \leq f(s)\) (minimisation) or \(f(s^*) \geq f(s)\) (maximisation) for all \(s \in S\).

How do we formulate a problem for a randomised search heuristics?

1. **Problem representation**
   Encoding of candidate solutions

2. **Objective**
   Optimisation goal

3. **Evaluation function**
   Indicates the quality of a given candidate solution
Problem Representation

Genotype-Phenotype Mapping

Phenotype Space $P$ (Solution Space) \hspace{2cm} \text{Decoding} \hspace{2cm} \text{Genotype Space } S \text{ (Search Space)}

$h: S \rightarrow P$
Problem Representation

Genotype-Phenotype Mapping

Evaluation Function

Decoding

Search Operators

**Phenotype Space** $P$  
(Solution Space)

**Genotype Space** $S$  
(Search Space)

$x = 010101$

$S = \{0, 1\}^n$

Genotype-Phenotype Mapping

**Problem Representation**
Example: Vertex Cover Problem

Definition

Given: unweighted, undirected graph \( G = (V, E) \)
Goal: Find a minimum set of vertices \( V' \subset V \) such that each edge is covered at least once, i.e.,
\[
\forall \{u, v\} \in E: \{u, v\} \cap V' \neq \emptyset
\]

Is this a feasible solution? No!
Example: Vertex Cover Problem

Definition

Given: unweighted, undirected graph $G = (V, E)$

Goal: Find a minimum set of vertices $V' \subseteq V$ such that each edge is covered at least once, i.e.,

$$\forall \{u, v\} \in E: \{u, v\} \cap V' \neq \emptyset$$

Optimal solution:
Example: Vertex Cover Problem (cont.)

**Definition**

<table>
<thead>
<tr>
<th>Given</th>
<th>unweighted, undirected graph $G = (V, E)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal</td>
<td>Find a <strong>minimum</strong> set of vertices $V' \subset V$ such that each edge is covered at least once, i.e., $\forall {u, v} \in E: {u, v} \cap V' \neq \emptyset$</td>
</tr>
</tbody>
</table>

A straightforward problem formulation

1. **Problem representation**

   $S = \{0, 1\}^{|V|}$

   How do we map nodes to the bits in a bit string $x \in S$?

   $|V|!$ different mappings $m: V \rightarrow \{1, \ldots, |V|\}$.

   **Genotype-phenotype mapping:** $h(x) = \{v_{m(i)} \in V \mid x[i] = 1\}$

2. **Objective**

   Find a minimum vertex cover.

3. **Evaluation function**

   $$f(x) = \begin{cases} |V'| & \text{if } V' \text{ is feasible} \\ (|V| + 1) \cdot |\{e \in E \mid e \cap V' = \emptyset\}| & \text{otherwise} \end{cases}$$
Standard Search Spaces

Binary Genotypes \( S = \{0, 1\}^n \)

Examples: Pseudo-Boolean function optimisation \((f: \{0, 1\}^n \to \mathbb{R})\), combinatorial problems such as Vertex Cover, SAT or Knapsack

Continuous Genotypes \( S = \mathbb{R}^n \)

Examples: Continuous function optimisation \((f: \mathbb{R}^n \to \mathbb{R})\), real-valued parameter estimation

Permutation Genotypes \( S = S_n \)

Examples: Travelling Salesperson Problem, Knapsack Problem

Remark Genotype-phenotype mapping allows us to switch between different representations, e.g., use of a continuous genotype for permutation problems.
References and Further Reading

*Analyzing Evolutionary Algorithms: The Computer Science Perspective*, chapter 2.5.  


*Design of Modern Heuristics*, chapter 4.  