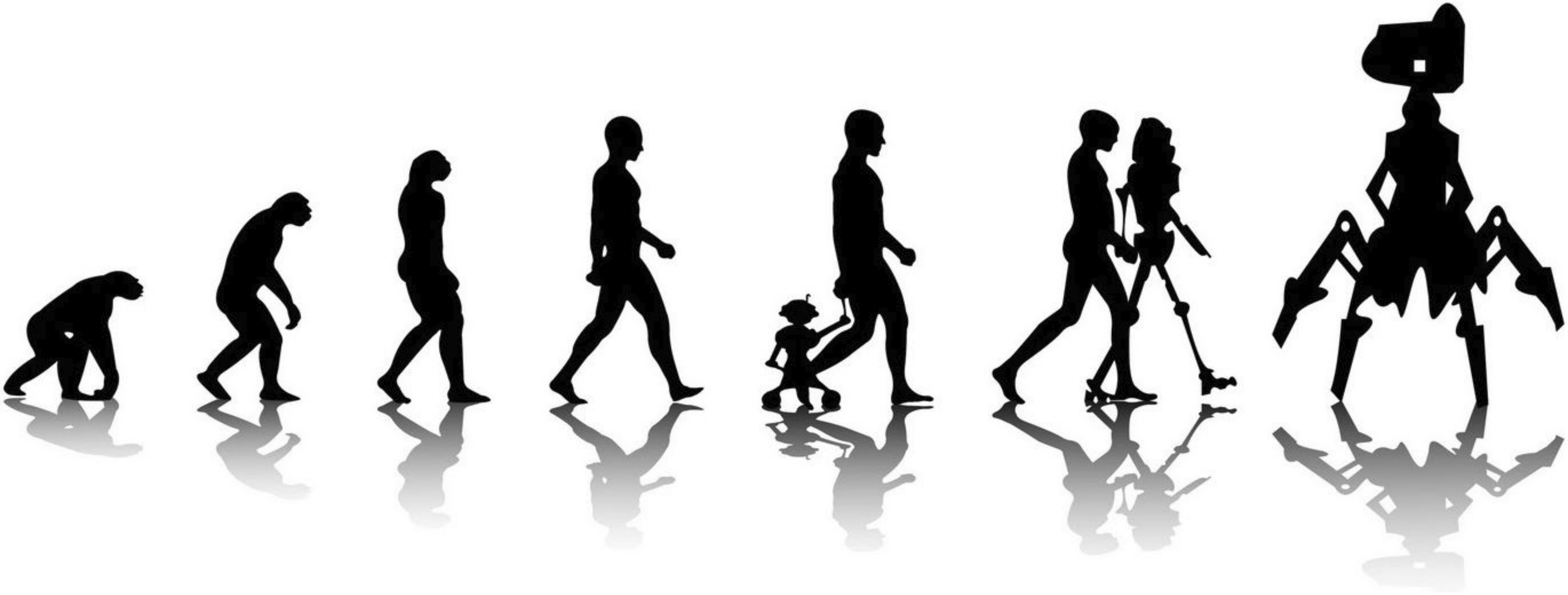


Modern Robotics: Evolutionary Robotics

COSC 4560 / COSC 5560

Professor Cheney

4/9/18

COSC 4560: Project Proposal

Mike Schwindt
University of Wyoming
Laramie, Wyoming
mschwin1@uwyo.edu

Evolutionary Robotics, Neural Networks, & Sexual Reproduction

Lucas Nicodemus

Surrogate Models for Fitness Evaluations

Survey paper

Surrogate-assisted evolutionary computation: Recent advances and future challenges

Yaochu Jin

Department of Computing, University of Surrey, Guildford, Surrey, GU2 7XH, UK

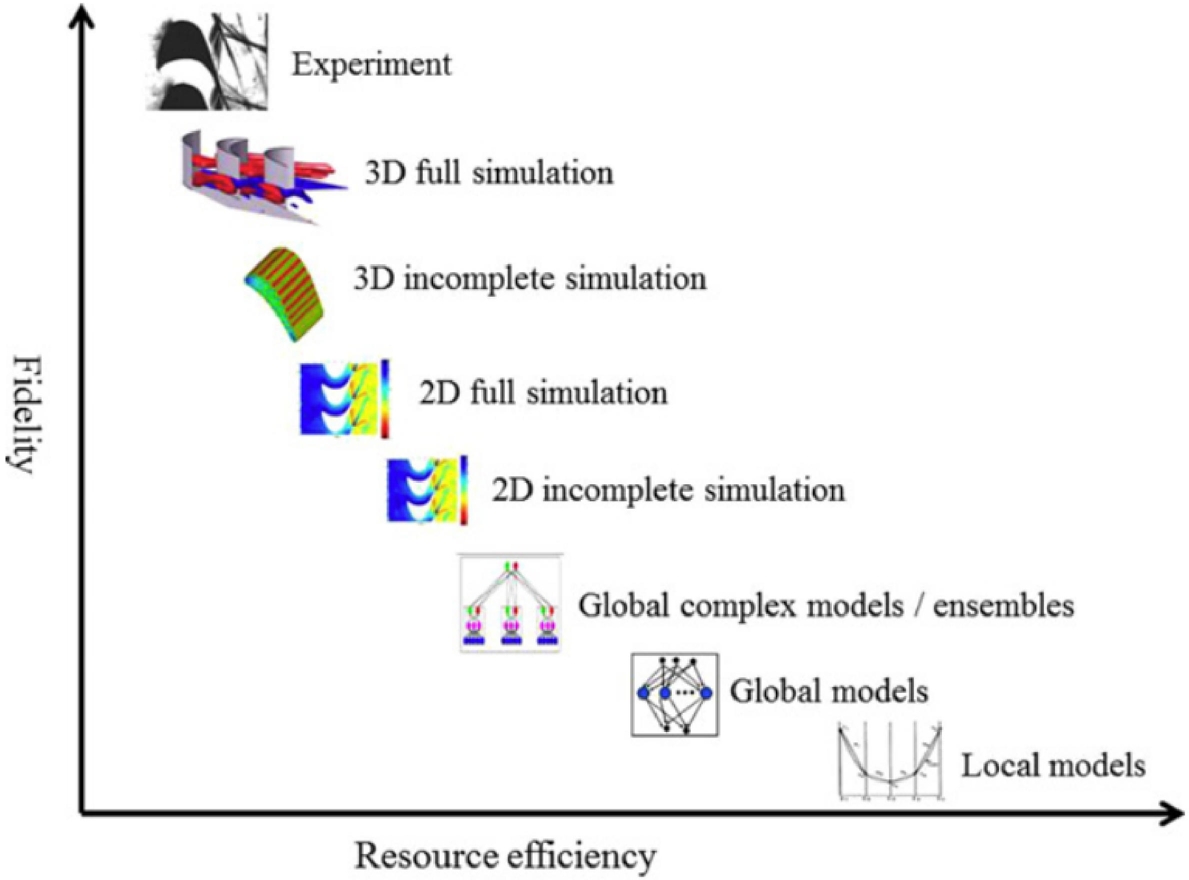


Fig. 1. An illustration of a trade-off between fidelity (approximation accuracy) and computational cost. Usually, high-fidelity fitness evaluations are more time-consuming. By contrast, low-fidelity fitness evaluations are often less time-consuming.

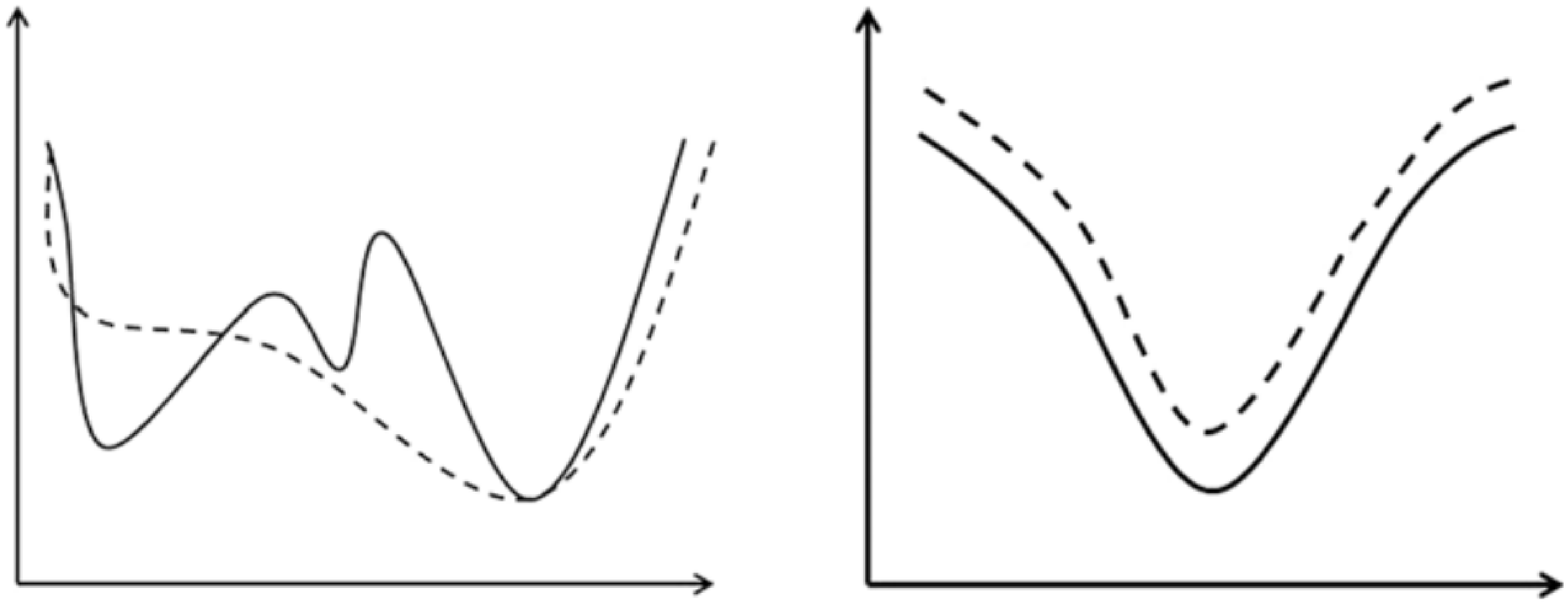
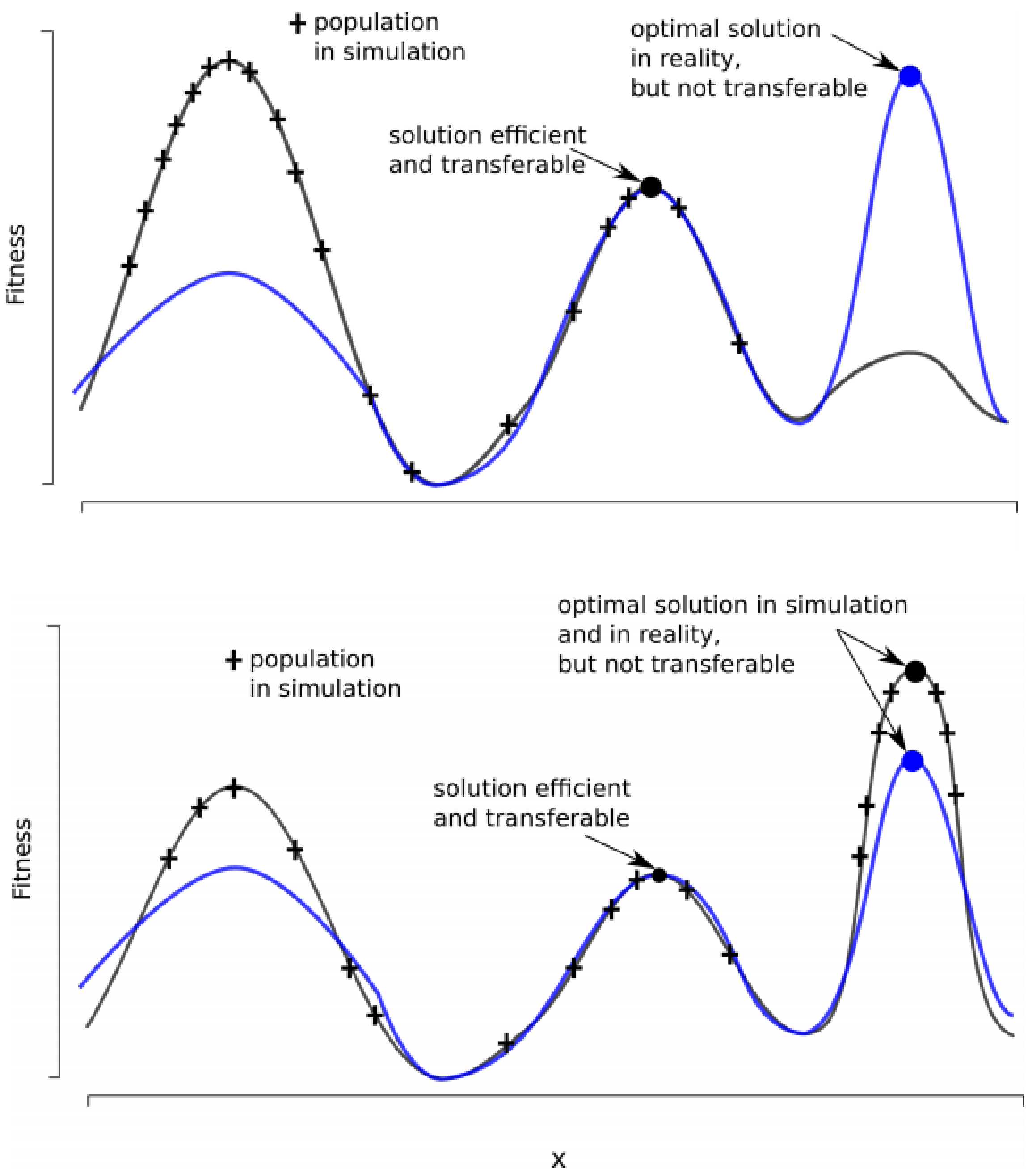
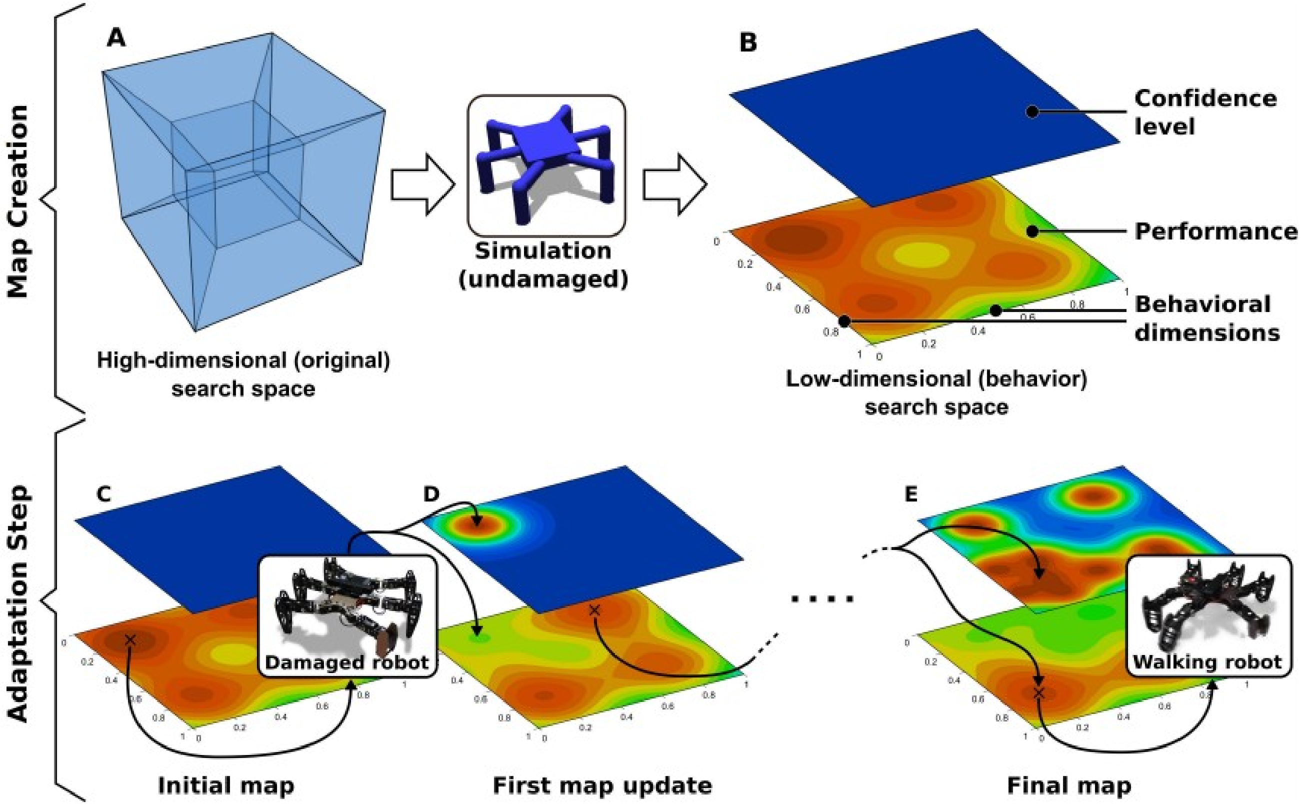
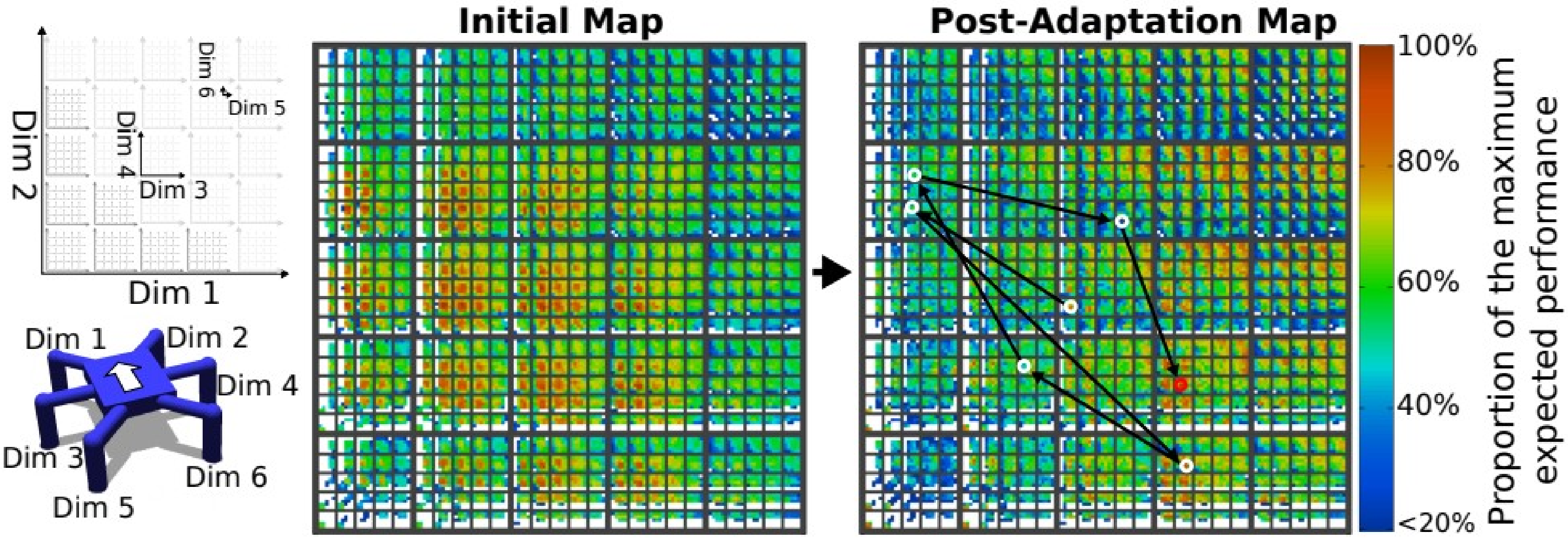


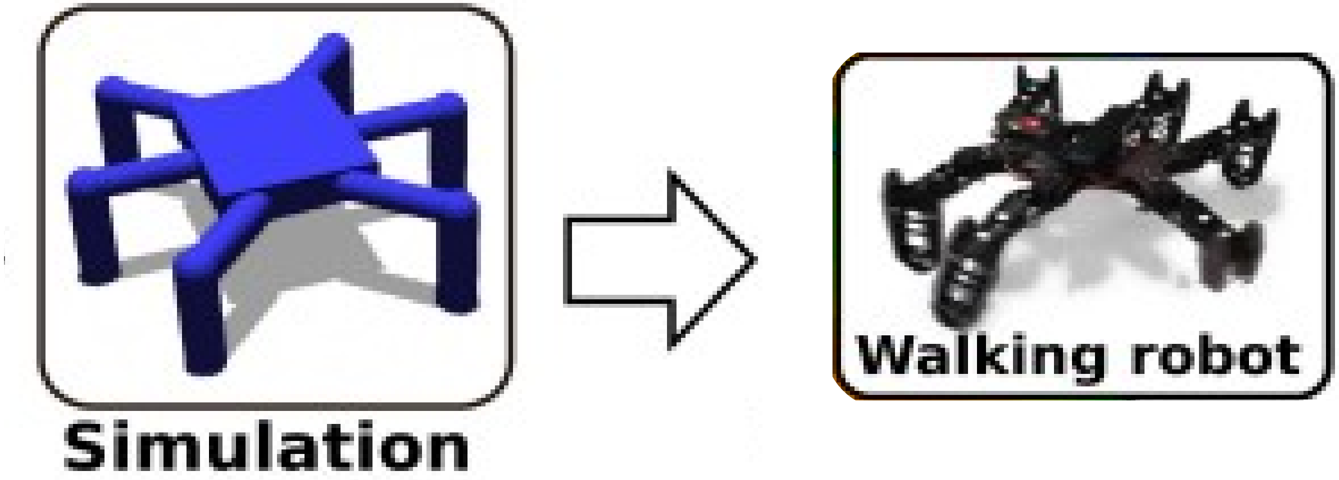
Fig. 4. Examples of surrogates that have a large approximation error but are adequately good for evolutionary search. Solid curves denote the original function and dashed curves are their approximation.



+ population in simulation
optimal solution in reality, but not transferable
solution efficient and transferable
Fitness
+ population in simulation
optimal solution in simulation and in reality, but not transferable
solution efficient and transferable
Fitness
x


Map Creation
A
High-dimensional (original) search space
Simulation (undamaged)
B
Confidence level
Performance
Behavioral dimensions
Low-dimensional (behavior) search space
Adaptation Step
C
D
E
Damaged robot
Walking robot
Initial map
First map update
Final map


Initial Map
Post-Adaptation Map
Dim 1
Dim 2
Dim 3
Dim 4
Dim 5
Dim 6
100%
80%
60%
40%
<20%
Proportion of the maximum expected performance



Simulation

Walking robot

Coevolution of Fitness Predictors

Michael D. Schmidt and Hod Lipson, *Member, IEEE*

1) Motivation: There are several reasons for utilizing fitness approximation through modeling.

- **Reducing complexity**: Many applications of evolutionary algorithms are in high-complexity or intractable domains, where the fitness calculation can be prohibitively time consuming. For example, fitness modeling has been applied to structural design optimization [1], [2], [21]–[25] that often requires time-consuming finite-element calculations. Often the resolution provided by the exact fitness objective is unnecessary for evolutionary progress.

- **No explicit fitness**: Many domains do not have a computable fitness. For example, in human interactive evolution [26] (e.g., evolution of art and music), a human user must select favorable individuals. Fitness models have been applied in these domains to reduce user fatigue and define a computable fitness landscape that can be searched, while waiting for the user to give more feedback [11], [27], [28].

- **Noisy fitness**: Some fitness functions are very noisy. To produce stable fitness rankings, algorithms typically average many evaluations, but this can greatly increase the computational cost [29]. An alternative approach may be to develop a statistical model [30].

- **Smoothing landscapes**: Almost all evolutionary domains suffer from multimodal landscapes that are often dense with local optima. Fitness approximation can greatly reduce the frequency and severity of local optima. Landscape smoothing has been observed with interpolation, kernels, and fitness clustering [24], [25], [31], [32].

- **Promoting diversity**: When models smooth fitness landscapes, they often flatten local optima or produce different regions with similar fitness. While this is undesirable when using a single model throughout evolution, it can be advantageous for producing diversity as long as the fitness model continuously adapts, as is proposed in this paper.

Subsample Fitness Predictors

1) Fitness Predictor Encoding: Training data in symbolic regression typically consists of hundreds to thousands of data points (e.g., experimental measurements) providing output values for a sample of inputs. In our symbolic regression experiments, the fitness predictor is a small subset of these points. Instead of measuring the exact objective fitness of candidate solutions, a subjective fitness is obtained by measuring the error on the select handful of data points of a given fitness predictor.

The fitness predictor is encoded as a small array of indexes to the full training data set (size discussed in the next section). Each index in the predictor's array is free to reference any points in the training data examples and can repeatedly sample point if it likes (thus over emphasizing an area). The predicted fitness is calculated as

$$\text{predicted\_fitness}(s) = \frac{1}{n}\sum_{i=1}^{n}|s(x_i) - y_i|$$

where s is a candidate solution (algebraic expression), x_i and y_i are training data inputs and outputs in the training dataset indexed by the predictor, and n is the number of samples in the predictor.