A Unified Model of Optimisation Problems

Cristina C. Vieira
CSI – Centro de Sistemas Inteligentes
Universidade do Algarve
8005-139 Faro – Portugal
cvieira@ualg.pt

Carlos M. Fonseca
CSI – Centro de Sistemas Inteligentes
Universidade do Algarve
8005-139 Faro – Portugal
cmfonsec@ualg.pt

ABSTRACT

In this work, a conceptual software model of optimisation problems is developed. Problem-specific aspects are clearly identified as such. To achieve the desired separation between problems and solvers, the details of the problem are encapsulated, and *mechanisms* capable of supporting the optimisation process are provided in a problem-independent way, allowing optimisers to be formulated at a more abstract level. The proposed model has been prototyped in Python.

Categories and Subject Descriptors

I.6.4 [**Simulation and Modeling**]: Model Development; D.2.1 [**Software Engineering**]: Requirements/Specifications

General Terms

Design

Keywords

Optimisation problems, Conceptual software model

1. INTRODUCTION

Current software environments for optimisation are generally centred on given families of optimisation methods. Unfortunately, problem-specific and solver-specific aspects are not always clearly separated. A clearer separation between them would improve the understanding of the optimisers themselves, facilitate deployment, and enable fairer assessment and comparison of the performance of different optimisation methods.

2. PROBLEM ANALYSIS

To achieve a greater separation between problem-specific and solver-specific aspects, it is important to recognise that an optimisation problem exists independently of any method which may be used to solve it. Although optimisation problems are usually formulated in terms of numerical functions, a more general definition may be given:

Definition 1 (Optimisation problem). *An optimisation problem is an abstract problem [1] where each instance in I is a* pre-order $\preceq$ *on the set of problem solutions S. An element of S is a solution of a problem instance in I if and only if it is a* minimal *element [2] of S under the corresponding pre-order (alternatively,* maximal*).*

GECCO'07, July 7–11, 2007, London, England, United Kingdom.
ACM 978-1-59593-697-4/07/0007.

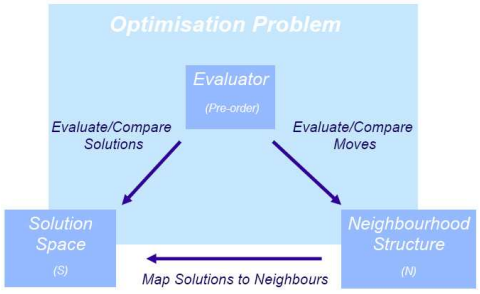

Figure 1: Conceptual model

This general definition encompasses a number of different optimisation scenarios, namely, local and global, discrete and continuous, single-objective and multi-objective optimisation. In practice, global optimisation problems are usually very difficult to solve directly, and *local* optimisation problems are often considered instead.

Definition 2 (Local optimisation problem). *A local optimisation problem is an optimisation problem such that an element $s \in S$ is a solution of a problem instance in I if and only if it is a* minimal *element of a subset $N(s) \subseteq S$, under the pre-order defined by that instance. The function $N(s) : S \rightarrow 2^S$ is called a* neighbourhood function*, and endows S with a* neighbourhood structure*.*

These definitions suggest the decomposition of the problem specification into three inter-related entities: problem instances, solution spaces and neighbourhood structures.

3. THE CONCEPTUAL SOFTWARE MODEL

Having identified the relevant entities and the corresponding relationships in the mathematical formulation of optimisation problems, a conceptual model for their implementation in software is depicted in Figure 1.

4. REFERENCES

[1] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* MIT Press, Massachusetts, 1990.

[2] P. Taylor. *Practical Foudations of Mathematics.* Cambridge University Press, 1999.