Towards Constraint-Based Local Search for Automatic Test Data Generation

Nadjib Laazaar*
Arnaud Gotlieb* Yahia Lebbab**
* IRISA-INRIA, Campus Beaulieu, Rennes, France
** Université d’Oran Es-Senia, Oran, Algeria

Local Search

- LS is a metaheuristic for solving computationally hard optimization problems.
- LS algorithm moves from solution to solution in the search space until a near optimal solution is found or a time bound is elapsed.

Constraint-Based Local Search

CBLS, as introduced by Michel and Van Hentenryck [VPH MIT Press 2005], aims at introducing local search algo.
within constraint-based models.

It consists of:

A modeling component offering several concepts like invariants and differentiable objects.

A search component aiming at simplifying the implementation of heuristics and meta-heuristics.

Our Approach

In our previous work, we built a Constraint-Based Testing environment that uses static single assignment as the constraint generation step and domain reduction as the constraint solving step to the structural testing for C program [Gotlieb et al. ISSSTA’98].

SSA Form

Each use of a variable refers to a single definition

\[ x := x + y; \]
\[ y := y + x; \]
\[ y := x + y; \]

\[ \phi \text{ function } \]

Constraint Solving

- Constraint Filtering
- Variable Labeling
- Constraint propagation

Euclidean

Implemented in Sicstus Prolog in the Euclide tool

SSA For Structured Abstract Syntax trees: algo. From [Brandis & Møssenbock TOPLAS’94]

Points-to analysis based on set-based data structures

Architecture:

Current Work

Our CBT framework implements arithmetic constraints, logical constraints and special combinator to modelize arrays, pointers and control structures like ITE(C, THEN, ELSE).

[Gotlieb et al. IST’06]

For each of these constraints, the reflexive interface shown at side must be implemented.

Thanks to this implementation, exploiting and experimenting various local search algorithms will become easier.

Research works remain to decide how to implement the cost function that minimize the number of conflicts in the presence of complex control structure.

An Example

uch getmid(uch x, uch x, uch x) { 1. uch mid; 2. mid = x; 3. if(x < x) 4. if(x > x) 5. mid = x; 6. else if(x < x) 7. mid = x; 8. else if(x > x) 9. mid = x; 10. else if(x = x) 11. mid = x; 12. return mid; }

Solution!!

\[ X = \{ x_1, x_2, x_3 \} \]
\[ D(x_1) = D(x_2) = D(x_3) = 0.255 \]

C

GSP

Abstract class Constraint {
bool isTrue(); 1. getName(); 2. int violations(size); 3. int violations(name); 4. getAssignDts(name, x, v); 5. getAssignDts(name, v, x); 6. interface for differentiable objects

[Diagram showing the graph and solution]