Evolutionary Algorithms and Constraint Satisfaction: Definition, Survey, Methodology, and Research Directions

4. Evolutionary CSP solvers

-Eiben et al.

- Combination of traditional CSP solving heuristics with genetic algorithm.
- The heuristics can be incorporated into both mutation and crossover operator.

-Paredis:

- Putting the constraints into a second population and co-evolving them with the population of candidate solutions.
- The contents of the constraint population does not change and fitness function is used only for set the probability the constraint is selected.
- It works with a fitness function that is changing over time.

-Dozier:

- ASGA:Heuristic-based microgenetic approach.
- Use heuristics in the mutation and the crossover operators.
- On-line detection and exploitation of 'nogoods'(bad values for pair of variables) which changes fitness function adaptively.
- Arc-revision, Arc-inconsistency check:Enables GA to "realize when to quit"

-Riff-Rojas:

- Using knowledge on the constraint network to assist evolutionary search.
- 'Deep' heuristics in the sense that the algorithm look further than one constraint and take a part of the constraint network into account.
- Defines heuristic quality estimates of candidate solutions different from counting constraint violations or wrong variable instantiations.

-Marchiori:

- Introduced a method for solving binary CSPs given in a specific format.
- Preprocessing the problem before applying a GA to solve the problem.
- The GA, using a penalty function counting violated constraints, features a specific repair heuristic called dependency propagation.

-Eiben and van der Hauw

- SAW(stepwise adoptation of weights)-general mechanism to handle constraints.
- Idea:the constraints that are not satisfied after a certain number of steps must be hard and must be given a high weight(penalty)
- Adaptive fitness function.

5.Lessons Learned

-EA worked well in CSP solving though EAs are intrinsically unsuited for constrained problems.

-How can EAs work well then?

- Using heuristics in EA components; fitness function,operators, selection mechanism, repair procedure.
- Using adaptive features, in particular a fitness function that is being modified during a run.

-'Tricks'

- The presence of constraints facilitates measures on sub-individual structures.
- The composite nature of the problem leads to a composite evaluation function.

-Promising options

- Use heuristics to estimate the sub-individual entities.
- Exploit the composite nature of the fitness function and change its composition over time.
- Try small populations and mutation-only schemes

6. Methodological Considerations

-Methodoligical questions

- Which benchmarks problems and problem instances should be used?
  - *Repository approach:use prepared problem instances
  - *Generator approach:instances are generated on-the-spot.
- Which competitor algorthms should be used?
- Which comparative measures should be used?
  - *Speed:CPU time, user time, Average number of Evaluations to a Solution
  - *Solution quality:Success rate, Mean Best Fitness
  - *Proability of finding a solution

7.Research Directions

- Study of the problem area. A lot can be learned from the traditional constrained literature about such problems. Existing knowledge should be imported into core EC research.
- Cross-fertilization between the insights concerning EAs for COPs and CSPs. At present, these two sub-areas are practically unrelated.
- Sound methodology:how to set up fair experimental research, how to obtain good benchmarks, how to compare EAs with other techniques.
- Theory:better analysis of the specific features of constrained problems, and the influence of these features on EA behavior.