Problem Solving and Search in Artificial Intelligence

Algorithm Configuration

Nysret Musliu
Database and Artificial Intelligence Group,
Institut of Logic and Computation, TU Wien

Motivation

- Metaheuristic techniques usually include several parameters
  - Tabu search: length of tabu list, type of memory, …
  - Simulated annealing: start and end temperature, decrease of temperature…
  - Iterated local search: size of perturbation, acceptance criteria, running time of local search procedure …
  - Evolutionary algorithms: population size, crossover rate, mutation rate,…
  - …
- Different components can be used
  - Neighborhood structure
  - Mutation type/crossover type
  - …

Finding appropriate parameters/components to be used is crucial for the performance of heuristics

Algorithm configuration (setting of parameters)

- Parameters are determined manually
- Automated algorithm configuration
  - Off-line parameter setting
  - On-line parameter setting

Manual Algorithm Configuration (I)

- Select different configuration for parameters
- Select representative instances for the problem to be solved
- Run experiments on instances with each parameter configuration (typically many runs per instance should be executed)
- Select the best configuration based on quality of solutions, running time of algorithm, …
- Statistical Analysis

Manual Algorithm Configuration (II)

- Disadvantages of manual configuration
  - Time consuming for the designer of algorithms
  - Limited number of configuration can be tested
  - Hard to find the best configuration

  - Alternatives:
    - Automated algorithm configuration

Automated Algorithm Configuration

- Input for off-line configuration problem:
  - Algorithm A
  - A set of parameter configurations
  - A set of input instances
- Problem:
  - Find parameter configuration that gives the best results on the input instances (e.g. solutions with best quality, time performance, …)
- This problem is a search problem - > Number of solutions is equal to number of possible parameter configuration

Off-line algorithm configuration examples (I)

ParamILS [3]:

- Applies iterated local search to find the best configuration
  - Search space: all possible parameter configurations
  - Objective function: the performance of the algorithm with a specific configuration
  - Neighborhood: modification of one parameter value at a time
  - Additional mechanism to speed-up the algorithm (by avoiding unnecessary runs)
- Applied for configuration of CPLEX, SAT algorithms, …

[3] Frank Hutter, Holger H. Hoos, Kevin Leyton-Brown, Thomas Stützle: ParamILS: An Automatic Algorithm Configuration Framework. J. Artif. Intell. Res. (JAIR) 36: 267-306 (2009) (http://www.cs.ubc.ca/labs/beta/Projects/ParamILS/)

Off-line algorithm configuration examples (II)

Iterated Race (irace): (Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Thomas Stützle, and Mauro Birattari. **The irace package, Iterated Race for Automatic Algorithm Configuration**. Technical Report TR/IRIDIA/2011-004, IRIDIA, Université libre de Bruxelles, Belgium, 2011.)

Three steps:

- Sampling new configurations according to a particular distribution
- Selecting the best configurations from the newly sampled ones by means of racing
- Updating the sampling distribution in order to bias the sampling towards the best configurations

Off-line algorithm configuration examples (III)

GGA - A Gender-Based Genetic Algorithm [4]

- Introduces a gender separation
- Speedup with parallelization

ISAC - Instance-Specific Algorithm Configuration [5]

- Integrates GGA and stochastic offline programming
- Training instances are clustered based on some features
- Best parameters are found for each cluster with GGA
- Offers selection of best parameters based on features of an input instance
- Applied for Set Cover, SAT and Mixed Integer Programming

[4] Carlos Ansótegui, Meinolf Sellmann, Kevin Tierney: A Gender-Based Genetic Algorithm for the Automatic Configuration of Algorithms. CP 2009: 142-157

[5] Serdar Kadioglu, Yuri Malitsky, Meinolf Sellmann, Kevin Tierney: ISAC - Instance-Specific Algorithm Configuration. ECAI 2010: 751-756

Off-line algorithm configuration examples (IV)

SMAC [HHL2011]

http://www.cs.ubc.ca/labs/beta/Projects/SMAC/papers/11-LION5-SMAC-slides.pdf

[HHL2011] Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. Sequential Model-Based Optimization for General Algorithm Configuration. In LION-5, 2011.

http://www.cs.ubc.ca/labs/beta/Projects/SMAC/
http://www.ml4aad.org/algorithm-configuration/smac/

On-line parameter setting

- Parameters change based on feedback during the search
- A parameter can change based on simple rules:
  - Increase tabu length or mutation size if diversification is needed
  - Apply neighborhood relations that imporved most of the times solutions
  - …
- More sophisticated techniques use machine learning techniques (for example reinforcement learning)

Adaptive techniques: Example I

Reactive tabu search [6]

- The prohibition T is determined based on feedback during the search
- T=1 in the start of the search
- T increases for 1 if diversification is needed
- The evidence that diversification is needed appears if for example
  - Previous solutions are repeated
  - The solutions have a short distance to the previous solutions
- T decreases if diversification is not needed

[6] Roberto Battiti, Giampietro Tecchiolli: The Reactive Tabu Search. INFORMS Journal on Computing 6(2): 126-140 (1994)

Adaptive techniques: Example II

Application of Reinforcement Learning (RL)

Online Control of Evolutionary Algorithms [7]

- A reinforcement method runs simultaneously with the evolutionary algorithm
- EA parameters: population size, tournament proportion, mutation probability, crossover probability
- RL learning changes above during the search based on the progress (best fitness, mean fitness, standard deviation, …) of EA between two time points

[7] A. E. Eiben, Mark Horvath, Wojtek Kowalczyk, Martijn C. Schut: Reinforcement Learning for Online Control of Evolutionary Algorithms. ESOA 2006: 151-160