Sim-EDA: A Multipopulation Estimation of Distribution Algorithm Based on Problem Similarity

Krzysztof Michalak[1)]

[1)] Wroclaw University of Economics, Poland [krzysztof.michalak@ue.wroc.pl]

The Sim-EDA algorithm

- A multipopulation algorithm…
- …but each subpopulation tackles a slightly different instance of the problem
- Island model + migration
- Migration influenced by similarities of problem instances
- Each subpopulation processed using an Estimation of Distribution Algorithm

- Main steps
  1. Initialize probabilistic models
  2. The main loop
     - 2.1. Generate new specimens
     - 2.2. Select source populations and specimens for migration
     - 2.3. Migrate specimens
     - 2.4. Update probabilistic models
- Details: proceedings, page 239

Problem similarity

- Problem instance similarity matrix:

$$S_{[N_{prob} \times N_{prob}]}$$

where:

N_{prob} – the number of problem instances

higher values = more similar instances

- Similarity between the *i*-th and *j*-th problem instances is expressed numerically as $S_{i,j}$
- In this paper problem instances are represented as weighted graphs
- The *i*-th problem instance has a cost matrix:

$$C^{(i)}_{[N \times N]}$$

- Similarity between *i*-th and *j*-th problem instance:

$$S_{i,j} = -\sum_{p=1}^{N}\sum_{q=1}^{N}(C^{(i)}_{p,q} - C^{(j)}_{p,q})^2.$$

Migration

- For each destination population *i* select the source population *j* based on values of $S_{i,j}$
- Migration strategies:
  - Nearest – Migrate specimens from the most similar subproblem (largest $S_{i,j}$ for $j \neq i$)
  - Rank - Rank all subopulations from the largest $S_{i,j}$ to the smallest (for $j \neq i$) and apply roulette-wheel selection
  - Uniform – Migrate specimens from a subproblem chosen at random with uniform probability (from $j \neq i$)
  - None – No migration
- Incoming specimens are added one by one
  - *x* – incoming specimen
  - *w* – the weakest specimen in target population
  - binary tournament between *x* and *w*
  - if *x* wins it replaces *w*

Test Problems – The Max-Cut

- Find a partition of a graph $G = \langle V, E \rangle$ with *N* vertices

$$V' \cup V'' = V \qquad V' \cap V'' = \emptyset$$

- Lengths of edges given in a cost matrix
- Each solution is a binary vector
- Minimization problem:

$$\text{maximize } f(V') = \sum_{i \in V', j \in (V - V')} c_{ij},$$

$$\text{subject to } V' \in 2^V,$$

Test Problems – The TSP

- Find the shortest route through *N* cities, visiting each city only once
- Distances given in a cost matrix
- Each solution is a permutation
- Minimization problem:

$$\text{minimize } f(\pi) = C_{\pi(N)\pi(1)} + \sum_{i=1}^{N-1} C_{\pi(i)\pi(i+1)}$$

$$\text{subject to } \pi \in \Pi(N)$$

where:

$\Pi(N)$ - the set of all permutations of numbers 1, …, *N*

Generating similar problem instances

- $N_{sub} = 20$ cost matrices:

$$C^{(1)}_{[N \times N]}, \ldots, C^{(20)}_{[N \times N]}$$

- In $C^{(1)}_{[N \times N]}$ the elements were drawn from U[0, 100]
- $C^{(j)}_{[K \times K]}$ was generated from $C^{(j-1)}_{[K \times K]}$ by replacing 1 / N_{prob} (1/20 = 5%) of elements by random values
- To ensure symmetry the elements above and below the diagonal kept identical

Problem-specific EDAs

The Max-Cut

- Model: a vector $P \in [0, 1]^N$
  - p_i – how probable it is that *i*-th element equals 1?
- Initially: P = [0.5, 0.5, …, 0.5]
- Model update from Population-Based Incremental Learning (PBIL) is used

The TSP

- Model: a matrix $Q_{[N \times N]}$ with $q_{i,j} \in [0, 1]^N$
  - $q_{i,j}$ – how probable it is that *j* is placed right after *i*
- Initially:

$$\mathbb{Q}_{i,j} = \begin{cases} 0 & \text{for } i = j \\ \frac{1}{N-1} & \text{for } i \neq j \end{cases}$$

- Model update from Population-Based Incremental Learning (PBIL) is used
- Best and worst permutations converted to Q_{best} and Q_{worst} by setting 1 iff *i* follows *j* in the permutation
- Obtaining permutations from $Q_{[N \times N]}$:

Algorithm 2. Initialization of a genotype for a new specimen for the TSP based on the probabilistic model $\mathbb{Q}$.

IN: $\mathbb{Q}$ - the probabilistic model
N - genotype length

OUT: X - a new genotype

$X[1] = \text{UniformSelection}(\{1, \ldots, N\})$
for $k = 2, \ldots, N$ **do**
  $U = \{1, \ldots, N\} - \{X[1], \ldots, X[k-1]\}$
  $i = X[k-1]$
  $S = \sum_{j \in U} \mathbb{Q}[i, j]$
  if $S > 0$ **then**
    $X[k] = \text{RouletteWheelSelection}(U, \{\mathbb{Q}[i, j]\}_{j \in U})$
  else
    $X[k] = \text{UniformSelection}(U)$
  end if
end for

Experiments and Results

- Aimed at comparing the migration strategies
- Problem instances:
  - The Max-Cut Problem with *N* = 12, 25, 50, 100, 250 and 500 nodes
  - The Travelling Salesman Problem (TSP) with *N* = 12, 25, 50, 100, 150, 200 and 250 cities
- 30 runs for each instance and migration strategy
- Results compared at the common time limit (a time in which the None strategy completed 200 generations)
- Comparison performed using the Wilcoxon statistical test
- Max-Cut results
  - Nearest vs. None: N = 12: Insignificant; N > 12: Significant
  - Rank & Uniform vs. None: N ≤ 100: Significant; N ≥ 250: Worse
- TSP results:
  - Nearest vs. None: N ≤ 50: Insignificant; N ≥ 100: Significant
  - Rank & Uniform vs. None: In most cases Insignificant
- The strategy of migrating specimens solely from the nearest subpopulation
  - Consistently produced better results than those obtained when no migration was used for larger instances
  - Was not significantly different from no migration for some smaller instances

RESEARCH POSTER PRESENTATION DESIGN © 2015
www.PosterPresentations.com