COMPARISON OF DECISIONS QUALITY OF HEURISTIC METHODS WITH SEQUENTIAL FORMATION OF THE DECISION IN THE GRAPH SHORTEST PATH PROBLEM

Eduard I. Vatutin

BOINC:FAST’17
Petrozavodsk, 2017
Classification of methods for solving discrete combinatorial optimization problems

Universal methods:
- Brute Force (branches and bounds strategy);
- Limited Depth First Search;
- Greedy approach;
- (Weighted) Random Search;
- Ant Colony Optimization;
- Simulated Annealing;
- Genetic Algorithms.

Special methods:
- Parallel-Sequential Method;
- Dijkstra’s algorithm;
- Hungarian algorithm (Kuhn-Munkers algorithm);
- ...
Classification of methods for solving discrete combinatorial optimization problems

Precise methods (ensures an optimal solution):
- Brute Force (branches and bounds strategy);
- Dijkstra’s algorithm;
- Hungarian algorithm (Kuhn-Munkers algorithm);
- Greedy approach (Minimal spanning tree).

Approximate methods (ensures an sub- or quasi-optimal solutions):
- Limited Depth First Search;
- Greedy approaches (most tasks);
- (Weighted) Random Search;
- Ant Colony Optimization;
- Simulated Annealing;
- Genetic Algorithms;
- Different special methods.
Classification of methods for solving discrete combinatorial optimization problems

Consecutive methods:
- Greedy approach;
- Some special methods (Dijkstra’s algorithm, parallel-sequential method, ...).

Iterative methods:
- Brute Force (branches and bounds strategy);
- Limited Depth First Search;
- (Weighted) Random Search;
- Ant Colony Optimization;
- Simulated Annealing;
- Genetic Algorithms.
Classification of methods for solving discrete combinatorial optimization problems

Limited-search methods:
- Brute Force, Branch and Bound strategy;
- Limited versions of Brute Force.

Methods with sequential formation of the decision:
- Greedy approach;
- Random search;
- Weighted Random search;
- Ant Colony optimization;
- Intelligent Water Drops method.

Methods based on modifying operations:
- Simulated Annealing;
- Genetic Algorithms;
- Bee Colony optimization method.

Methods based on the movement in arguments space:
- Particle Swarm Optimization;
- Firefly method;
- Fish School Search;
- Gravitational Search.
Assessment of the quality of decisions, convergence rate and computing time costs

Quality of decisions:
- during $C \to \infty$, $Q \to Q^*$;
- how to determine $C_{min}$ that $\Delta Q = |Q - Q^*|$ is acceptable for selected problem (for example, $\Delta Q/Q^*100\% < 5\%$)?

Ability of paralleling:
- Parallel execution is difficult or ineffective (for example, parallel-sequential method – theoretically no more than 10% speedup by Amdahl’s law);
- Weak or strong coupled implementations (AC with $M=1$ or $M=100$) – limiting classes of hardware;
- Trivial parallelized (random search, weighted random search).
Assessment of the quality of decisions based on samples of random source data

Sample of random source data
$$\Lambda = \{ G_1, G_2, ..., G_K \}$$

Average value of quality criteria
$$\bar{Q} = \frac{\sum_{i=1}^{K} Q(G_i)\phi(G_i)}{K}, \phi(G_i) \in \{0, 1\}$$
$$\bar{Q} < Q^* ?$$

Average deviation from optimum
$$\Delta \bar{Q} = \frac{\sum_{i=1}^{K} (Q(G_i) - Q^*(G_i))\phi(G_i)}{K}$$
Assessment of the quality of decisions based on samples of random source data

Probability of finding decision

\[ \bar{p} = \frac{\sum_{i=1}^{K} \Phi(G_i)}{K} \]

Probability of finding optimal decision

\[ \bar{p}_{opt} = \frac{\sum_{i=1}^{K} \theta(G_i)}{K}, \quad \theta(G_i) = \begin{cases} 0, & Q(G_i) > Q^*(G_i) \\ 1, & Q(G_i) = Q^*(G_i) \end{cases} \]

Average number of iterations (convergence rate)

\[ \bar{C} = \frac{\sum_{i=1}^{K} C(G_i)}{K} \]
Statement of a problem

\[ G = \langle A, V \rangle \]
\[ A = \{a_1, a_2, \ldots, a_N\} \]
\[ V = \{v_1, v_2, \ldots, v_M\}, A \times A \subseteq V \]
\[ d = \frac{M}{N(N-1)} \]
\[ l(v_i) = l_{j,k}, v_i = (a_i, a_k), \exists l(v_i) = \infty \]
\[ P = [a_{i_1}, a_{i_2}, \ldots, a_{i_m}] \]
\[ L = \sum_{j=1}^{m-1} l_{i_j, i_{j+1}} \rightarrow \min \]
Selecting the direction of movement in the combinatorial tree

\[ x = \arg \min_i F_i = \begin{cases} 
    f_i & \text{if } i \text{ arg min } \sum_{j=1}^{n} \eta_j \tau_j \end{cases} \]

\[ g(f_i, r_k) = \frac{r_k \eta_i \tau_i^{1-\alpha}}{\sum_{j=1}^{n} \eta_j \tau_j^{1-\alpha}} \rightarrow r_k \eta_i \tau_i^{1-\alpha} \]
Example of meta-optimization: weighted random search

\[ i = \arg \min_j \left[ L\left( a_{mek}, a_j \right)(1 + 2D(r_k - 0.5)) \right] \]

Average quality of the decision

Number of iterations
Deadlocks problem

\[ N = 30, d = 0.1 \]

<table>
<thead>
<tr>
<th>Метод</th>
<th>( \bar{L} )</th>
<th>( \Delta \bar{L} )</th>
<th>( \bar{p} )</th>
<th>( \bar{p}_{opt} )</th>
<th>( t, \text{мс} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>1,208</td>
<td>0,000</td>
<td>1,00</td>
<td>1,00</td>
<td>0,02</td>
</tr>
<tr>
<td>G</td>
<td>1,923</td>
<td>1,086</td>
<td>0,28</td>
<td>0,09</td>
<td>0,003</td>
</tr>
<tr>
<td>GR</td>
<td>3,706</td>
<td>2,497</td>
<td>1,00</td>
<td>0,12</td>
<td>0,01</td>
</tr>
<tr>
<td>RM</td>
<td>1,159</td>
<td>0,000</td>
<td>0,34</td>
<td>0,34</td>
<td>3,73</td>
</tr>
<tr>
<td>RMR</td>
<td>1,211</td>
<td>0,002</td>
<td>1,00</td>
<td>0,97</td>
<td>7,08</td>
</tr>
</tbody>
</table>
## Comparison of quality of decisions

### $N = 10$, $d = 0.5$

<table>
<thead>
<tr>
<th>Method</th>
<th>$L$</th>
<th>$\Delta L$</th>
<th>$\bar{p}$</th>
<th>$\bar{p}_{opt}$</th>
<th>$\bar{C}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>0,4864</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>1,1019</td>
<td>0,6322</td>
<td>0,8558</td>
<td>0,2810</td>
<td>1</td>
</tr>
<tr>
<td>RM</td>
<td>0,4821</td>
<td>0,0004</td>
<td>0,8535</td>
<td>0,8500</td>
<td>108</td>
</tr>
<tr>
<td>RMR</td>
<td>0,4866</td>
<td>0,0002</td>
<td>1</td>
<td>0,9975</td>
<td>108</td>
</tr>
<tr>
<td>WRM</td>
<td>0,4765</td>
<td>0,0010</td>
<td>0,8479</td>
<td>0,8436</td>
<td>22</td>
</tr>
<tr>
<td>WRMR</td>
<td>0,4873</td>
<td>0,0009</td>
<td>0,9998</td>
<td>0,9940</td>
<td>24</td>
</tr>
<tr>
<td>AC</td>
<td>0,4933</td>
<td>0,0069</td>
<td>0,9998</td>
<td>0,9504</td>
<td>11</td>
</tr>
<tr>
<td>ACR</td>
<td>0,4965</td>
<td>0,0101</td>
<td>0,9998</td>
<td>0,9367</td>
<td>10</td>
</tr>
<tr>
<td>UVLP</td>
<td>0,4964</td>
<td>0,0101</td>
<td>0,9997</td>
<td>0,9352</td>
<td>10</td>
</tr>
</tbody>
</table>

### $N = 50$, $d = 0.5$

<table>
<thead>
<tr>
<th>Method</th>
<th>$L$</th>
<th>$\Delta L$</th>
<th>$\bar{p}$</th>
<th>$\bar{p}_{opt}$</th>
<th>$\bar{C}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>0,1634</td>
<td>0</td>
<td>1</td>
<td>0,9975</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>1,5576</td>
<td>1,3952</td>
<td>0,9803</td>
<td>0,0839</td>
<td>1</td>
</tr>
<tr>
<td>RM</td>
<td>0,2876</td>
<td>0,1243</td>
<td>0,9787</td>
<td>0,2788</td>
<td>536</td>
</tr>
<tr>
<td>RMR</td>
<td>0,2874</td>
<td>0,1240</td>
<td>1</td>
<td>0,2853</td>
<td>536</td>
</tr>
<tr>
<td>WRM</td>
<td>0,1721</td>
<td>0,0089</td>
<td>0,9806</td>
<td>0,7892</td>
<td>197</td>
</tr>
<tr>
<td>WRMR</td>
<td>0,1727</td>
<td>0,0092</td>
<td>1</td>
<td>0,8023</td>
<td>200</td>
</tr>
<tr>
<td>AC</td>
<td>0,2060</td>
<td>0,0403</td>
<td>1</td>
<td>0,5429</td>
<td>25</td>
</tr>
<tr>
<td>ACR</td>
<td>0,2073</td>
<td>0,0421</td>
<td>1</td>
<td>0,5331</td>
<td>25</td>
</tr>
<tr>
<td>UVLP</td>
<td>0,2069</td>
<td>0,0436</td>
<td>1</td>
<td>0,5164</td>
<td>25</td>
</tr>
</tbody>
</table>

### $N = 10$, $d = 0.2$

<table>
<thead>
<tr>
<th>Method</th>
<th>$L$</th>
<th>$\Delta L$</th>
<th>$\bar{p}$</th>
<th>$\bar{p}_{opt}$</th>
<th>$\bar{C}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>O</td>
<td>1,0412</td>
<td>0</td>
<td>0,8179</td>
<td>0,8179</td>
<td>1</td>
</tr>
<tr>
<td>G</td>
<td>1,0577</td>
<td>0,2513</td>
<td>0,4052</td>
<td>0,2747</td>
<td>1</td>
</tr>
<tr>
<td>RM</td>
<td>0,9535</td>
<td>0,0629</td>
<td>0,4042</td>
<td>0,3291</td>
<td>73</td>
</tr>
<tr>
<td>RMR</td>
<td>1,0412</td>
<td>0,0000</td>
<td>0,8179</td>
<td>0,8179</td>
<td>73</td>
</tr>
<tr>
<td>WRM</td>
<td>0,8846</td>
<td>0,0422</td>
<td>0,4069</td>
<td>0,3485</td>
<td>8</td>
</tr>
<tr>
<td>WRMR</td>
<td>1,0423</td>
<td>0,0011</td>
<td>0,8179</td>
<td>0,8166</td>
<td>5</td>
</tr>
<tr>
<td>AC</td>
<td>1,0415</td>
<td>0,0003</td>
<td>0,8179</td>
<td>0,8166</td>
<td>7</td>
</tr>
<tr>
<td>ACR</td>
<td>1,0417</td>
<td>0,0004</td>
<td>0,8179</td>
<td>0,8170</td>
<td>3</td>
</tr>
<tr>
<td>UVLP</td>
<td>1,0436</td>
<td>0,0027</td>
<td>0,8173</td>
<td>0,8107</td>
<td>7</td>
</tr>
</tbody>
</table>

Differ results at differ values of N and d!
Experimental results.
Variations of ant colony optimization approach
Ant colony optimization (v1 и v2) with combinatorial returns

* – statistically equal
** – precise equal
Experimental results. Combinatorial returns strategy
Greedy approach

\[ N = \frac{A}{d} \]

\[ d = \frac{M}{N(N-1)} \]

\[ N = A \frac{N(N-1)}{M} \Rightarrow \]

\[ 1 = A \frac{N-1}{M} \Rightarrow \]

\[ M = A(N-1). \]
Random Search

$N$ $d$

$RM/RMR$

$RMR$

$0,0$ $0,1$ $0,2$
Weighted Random Search

![Graph showing Weighted Random Search (WRM) and Weighted Random Model (WRMR) with axes N and d.](image)
Ant colony optimization (v1 и v2)
Experimental results. Weighting heuristics within step by step construction of decision
Without combinatorial returns

\[ N \]

\[ \text{WRM} \]

\[ \text{RM/WRM} \]

\[ d \]

\[ 0 \]

\[ 0.0 \]

\[ 0.1 \]

\[ 0.2 \]

\[ 0,0 \]

\[ 0,1 \]

\[ 0,2 \]

\[ \text{RM} \rightarrow \text{WRM} \]

\[ \text{RM} \rightarrow \text{WRM} \rightarrow \text{AC} \]
With combinatorial returns

\[ \text{RMR} \rightarrow \text{WRMR} \]

\[ \text{RMR} \rightarrow \text{WRMR} \rightarrow \text{ACR} \]
Experimental results. Total comparison
Total comparison of all methods
New methods (from BOINC FAST 2015 :)
Random Walks

\[ x_n = x_0 + \sum_{i=1}^{n} r_i \]

\[ f^* = \min_{i=0, n} f(x_i) \]

\[ x^* = \arg \min_{i=0, n} f(x_i) \]

\[ P_n = o_i(P_{n-1}) \]
Modifying operations example

Source path:

\[ P = [ a_{i_1}, ..., a_{i_{R-1}}, a_{i_R}, a_{i_{R+1}}, ..., a_{i_Q} ] \]

Adding random vertex (operation 1)

\[ P' = [ a_{i_1}, ..., a_{i_R}, a_j, a_{i_{R+1}}, ..., a_{i_Q} ] \]

Deleting random vertex (operation 2)

\[ P' = [ a_{i_1}, ..., a_{i_{R-1}}, a_{i_{R+1}}, ..., a_{i_Q} ] \]

- more complex modifying operations can be developed
Particle Swarm Optimization

\[ X_i = \begin{bmatrix} x_1^{(i)} & x_2^{(i)} & \ldots & x_N^{(i)} \end{bmatrix} \]

\[ V_i = \begin{bmatrix} v_1^{(i)} & v_2^{(i)} & \ldots & v_N^{(i)} \end{bmatrix} \]

\[ X_i^{(t)} = X_i^{(t-1)} + V_i^{(t-1)} \]

\[ V_i^{(t)} = \alpha V_i^{(t-1)} + \beta r_k \otimes (X_i^* - X_i) + \gamma r_{k+1} \otimes (X^{**} - X_i) \]

\[ X \otimes Y = [x_1, x_2, \ldots, x_N] \otimes [y_1, y_2, \ldots, y_N] = [x_1y_1, x_2y_2, \ldots, x_Ny_N] \]

\[ X_i^* = \arg \min_{t=0,1} f(X_i^{(t)}) \]

\[ X^{**} = \arg \min_{i=1,2} f(X_i^*) \]
Particle Swarm Optimization: discrete problems

Sigmoid mapping:
\[
\sigma(y) = \frac{1}{1 + e^{-y}}
\]
\[
x = \begin{cases} 
1, & r_k < \sigma(y); \\
0, & \text{elsewhere.}
\end{cases}
\]

\[
x = \begin{cases} 
v_1, & 0 < r_k \sigma(y) \leq \frac{1}{Q}; \\
v_2, & \frac{1}{Q} < r_k \sigma(y) \leq \frac{2}{Q}; \\
\vdots \\
v_Q, & \frac{Q-1}{Q} < r_k \sigma(y) \leq 1,
\end{cases}
\]

\[
\mathbb{R}_1 \rightarrow \mathbb{R}_2
\]

Round mapping:
\[
i = \lfloor yQ \rfloor + 1
\]

Using modifying operations (without mapping):
\[
i = \lfloor \sigma(y)Q \rfloor + 1
\]
\[
X' = o_i(X)
\]
\[
d(X', Y) < d(X, Y)
\]

Program implementations:

- mapping probabilities \( p[\text{vertex\_num}][\text{path\_pos}] \) (PSO1, up to 700 MB RAM per workunit!);
- round mapping (PSO2);
- movement in the discrete space (PSO3).

\[
X = [2.03; 2.73; 6.72; 3.19; 1.62; 3.72; 4.26; 0.82] \implies
\]
\[
\implies P = \begin{bmatrix} a_{\text{beg}}, a_3, a_1, a_3, a_2, a_4, a_{\text{beg}} \end{bmatrix}_{\text{loop 1}}
\begin{bmatrix} a_{\text{beg}}, a_3, a_2, a_4, a_{\text{end}} \end{bmatrix}_{\text{loop 2}} \implies
\]
\[
P' = \begin{bmatrix} a_{\text{beg}}, a_3, a_2, a_4, a_{\text{end}} \end{bmatrix}
\]
Random Walks: results of experiments (PFOD)
Particle Swarm Optimization: results of experiments
Convergence rate analysis

The diagram illustrates the convergence rate analysis for different methods. The graph compares the performance of AC, ACR, AC2, GA, RS, BC, WRS, and SA. The x-axis represents the number of iterations ranging from 200 to 1000, while the y-axis indicates the convergence rate. The lines show how each method converges over time, with AC, ACR, AC2, and GA performing similarly and showing rapid convergence, while RS, BC, WRS, and SA converge more slowly.
Time costs analysis
Prospects for future research within Gerasim@home project

• adaptation and meta-optimization;
• pseudo triples of DLS (partial transversals + RS, WRS or AC);
• separation problem (LCSs): variations of consecutive and iteration methods;
• allocation problem (LCSs);
• combinatorial optimization problems (graphs theory, game theory, operations research).
The End.
Thanks!

The authors would like to thank all volunteers who took part in the calculation within the distributed computing project Gerasim@Home!

WWW: http://evatutin.narod.ru
E-mail: evatutin@rambler.ru
LJ: http://evatutin.livejournal.com
Skype: evatutin
vk, ok, facebook