Estimation of Distribution Algorithms for the Firefighter Problem

Krzysztof Michalak
Wroclaw University of Economics, Poland
krzysztof.michalak@ue.wroc.pl
Presentation Plan

- The Firefighter Problem (FFP)
- Estimation of Distribution Algorithms (EDAs)
- EDAs for the FFP
- Experiments
- Results
- Conclusions
The Firefighter Problem (FFP)

- Introduced by Hartnell in 1995\(^1\)
- Spread of fire is modelled on an undirected graph
  - Discrete-time model
  - Vertices are labelled:
    - 'B' – burning
    - 'D' – defended by firefighters
    - 'U' – untouched
  - Initially a certain number \(N_s\) of nodes are burning ('B') and the remaining ones are untouched

The Firefighter Problem (FFP)

- At each time step
  - $N_f$ firefighters are assigned to the untouched ('U') nodes. These nodes become defended ('D')
  - The fire spreads: nodes adjacent to the burning nodes catch on fire (unless they are defended by firefighters)

- The simulation stops when the fire is contained or when all undefended nodes are burning

- Representation of a solution:
  - A permutation = the order in which the nodes are protected
  - Other representations possible
The Firefighter Problem – An Example

\( N_s = 1 \)
\( N_f = 2 \)

\[ P = 3, 4, 2, 1, 6, 7, 8, 9, 10, 5 \]
\[ t = 0 \]

- \( N_s \) initial nodes are burning \( \bullet \)
The Firefighter Problem – An Example

\[ N_s = 1 \]

\[ N_f = 2 \]

\[ P = 3, 4, 2, 1, 6, 7, 8, 9, 10, 5 \]

\[ t = 1 \]

- \( N_f \) firefighters are assigned
The Firefighter Problem – An Example

\[ N_s = 1 \]
\[ N_f = 2 \]

\[ P = 3, 4, 2, 1, 6, 7, 8, 9, 10, 5 \]

\[ t = 1 \]
- the fire spreads to adjacent nodes

- the fire spreads to adjacent nodes
The Firefighter Problem – An Example

\[ N_s = 1 \]
\[ N_f = 2 \]

\[ P = 3, 4, 2, 1, 6, 7, 8, 9, 10, 5 \]
\[ t = 2 \]

- \( N_f \) firefighters are assigned
The Firefighter Problem – An Example

\[ N_s = 1 \]
\[ N_f = 2 \]

\[ P = 3, 4, 2, 1, 6, 7, 8, 9, 10, 5 \]
\[ t = 2 \]

- the fire spreads to adjacent nodes

Note, that at this point the fire cannot spread any further and the simulation can be stopped.
The Firefighter Problem (FFP)

- Task: Find the best assignment of firefighters to graph nodes

- Single-objective: save the highest possible number of nodes in the graph

- Multi-objective*: each node has several values assigned

- Single-objective with variable node cost:

\[ e(\pi) = \sum_{v \in V : l(v) \neq \emptyset} c_1(v) \]

where:

- \( c_1(v) \) - the cost assigned to the node \( v \).

Estimation of Distribution Algorithms

Main steps

- Select specimens from the population
- Update the probability model
- Draw a new population from the model
- Evaluate
Algorithm 1. A general structure of an EDA algorithm used in this paper.

IN: \( N_{\text{pop}} \) - The size of the population
\( N_{\text{sample}} \) - The size of a sample used for probabilistic model update

\( P := \text{InitPopulation}(N_{\text{pop}}) \)
\( M := \text{InitModel}() \)
\( B := \emptyset \)

while not StoppingCondition() do
  // Evaluation
  Evaluate(P)

  // Caching of the best specimen
  \( B := \text{GetBestSpecimens}(P, 1) \)

  // Update of the probabilistic model
  \( P_{\text{sample}} := \text{GetBestSpecimens}(P, N_{\text{sample}}) \)
  \( M := \text{UpdateModel}(P_{\text{sample}}, M) \)

  // New population
  \( P := \text{CreateNewSpecimens}(M, N_{\text{pop}} - 1) \)
  \( P := P \cup B \)
EDAs for the FFP

The State-Position (S-P) model

- Represents the relationship between
  - the graph state \( S \)
  - the position (number of the vertex \( v \)) at which a firefighter was assigned
  - the mean evaluation \( e \) which was finally achieved

- A graph state \( S \) is an element of \( L^{N_v} \), where:
  \[
  L = \{ 'B', 'D', 'U' \}
  \]
  \( N_v \) – the number of vertices
The **State-Position (S-P) model**

- The model is a list of triples:

\[ M = [\langle S_1, v_1, e_1 \rangle, \langle S_2, v_2, e_2 \rangle, \ldots, \langle S_n, v_n, e_n \rangle] \]

where \( S_i \in L^N, \quad v_i \in V \quad \text{and} \quad e_i \in \mathbb{R} \) for \( i = 1, \ldots, n \).

- An additional element:
  - A weight vector \( Q \)
  - \( Q[v] = \text{the sum of the reciprocals of the positions (counting from 1) of node } v \text{ in the solutions in } P_{\text{sample}} \)
  - Nodes used early in the solutions have higher weights
  - Used for selecting nodes to defend if no selection can be done based on the \( M \) model
EDAs for the FFP

- **The State-Position (S-P) model**
  - **Model $M$ update**
    - Take each permutation $\pi$ in the sample $P_{\text{sample}}$
    - Simulate spreading of fire using $\pi$ to assign firefighters
    - Each time a node $v$ is protected store a pair $<S, v>$, where $S$ is the current graph state
    - After the simulation finishes, calculate the final state evaluation $e$
    - Form a triple $<S, v, e>$ from each stored pair $<S, v>$
    - Average the evaluations for each distinct pair $<S, v>$

- **Update of the weight vector $Q$**
  - To obtain $Q[v]$ calculate the sum of the reciprocals of the positions (counting from 1) of node $v$ in the solutions in $P_{\text{sample}}$
EDAs for the FFP

The State-Position (S-P) model

Model sampling - simulate spreading of fire and at each time step $t$:

For each triple $<S, v, e>$ in the model $M$:

- Calculate the Hamming distance $h = H(S_t, S)$ between the current graph state $S_t$ and states $S$ stored in the triple

- Use one of the functions:
  - Linear: $f(x) = 1 + x$
  - Square: $f(x) = 1 + x^2$
  - Square root: $f(x) = 1 + \sqrt{x}$
  - Exponential: $f(x) = 3^x$

- Increase the weight $w[v]$ by $e / f(x)$

- If at least one $w[v] > 0$ perform roulette wheel selection w. r. t. $w[v]$

- Otherwise perform roulette wheel selection w. r. t. $Q[v]$
EDAs for the FFP

- The Mallows model
  - Exponential, unimodal distribution
  - Analogous to the Gaussian distribution
  - The probability distribution has two parameters:
    - A central permutation $\pi_0$
    - A spread parameter $\Theta$

- The Generalized Mallows model
  - Has $n - 1$ spread parameters

EDAs for the FFP

- **The EH-PBIL model**
  
  - Model used in the Edge Histogram-Based Sampling Algorithm (EHBSA)
  
  A matrix $\mathbb{P}_{[N \times N]}$ with $p_{i,j} \in [0, 1]^N$

  $$p_{i,j}$$ – how probable it is that $j$ is placed right after $i$ in good permutations

- An additional element:
  
  - A weight vector $W_s$
  
  Weights are used to determine with what probability each element can be used as the first one in the permutation
EDAs for the FFP

The EH-PBIL model

- **Weight vector $W_s$ update**
  - Take the first element $k$ in each solution in the sample $P_{sample}$
  - Increase $W_s[k]$ by the evaluation of the solution from which $k$ was taken

- **Probability matrix update**
  - Take the best solution $\pi^{(+)}$ and the worst one $\pi^{(-)}$ from the sample $P_{sample}$
  - Build a matrix $P^{(+)}$ from $\pi^{(+)}$ and $P^{(-)}$ from $\pi^{(-)}$
  - All elements in $P$ are set to zero except those elements $P_{i,j}$ for which $j$ immediately follows $i$ in $\pi$

\[ \pi = 3, 1, 4, 2 \]
EDAs for the FFP

- The **EH-PBIL model**
  - Update rule used in the PBIL algorithm
  - The positive learning rate $\eta^+$ and the negative learning rate $\eta^-$
  - Their sum is denoted $\eta = \eta^+ + \eta^-$
  - If $p_{ij}^{(-)} = p_{ij}^{(+)}$
    \[
    p_{ij} = p_{ij} \cdot (1 - \eta^+) + p_{ij}^{(+)} \cdot \eta^+
    \]
  - If $p_{ij}^{(-)} \neq p_{ij}^{(+)}$
    \[
    p_{ij} = p_{ij} \cdot (1 - \eta) + p_{ij}^{(+)} \cdot \eta
    \]
EDAs for the FFP

- The **EH-PBIL model**
  - The weight vector \( W_s \)
  - Motivation:
    - In the FFP the first element in the permutation is clearly identified
    - Some (non-uniformly-random) way of selecting this element is required
  - A different update rule than in the EHBSA
    - Preliminary results shown that PBIL model update is beneficial
  - Some results \( (N_v = 2500, \text{median from 50 runs}) \)

<table>
<thead>
<tr>
<th>( P_{\text{unif}} )</th>
<th>EHBSA Symmetric</th>
<th>EHBSA Asymmetric</th>
<th>PBIL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symmetric</td>
<td>6183</td>
<td>6201</td>
<td>6239</td>
</tr>
<tr>
<td>Asymmetric</td>
<td>6106</td>
<td>6106</td>
<td>6152</td>
</tr>
</tbody>
</table>

21
EDAs for the FFP

- The **EH-PBIL model**
  - Model sampling
    - The first element \((i = 1)\)
      - with a probability \(P_{\text{unif}}\) – select uniformly
      - with a probability \(1 - P_{\text{unif}}\) – select proportionally to elements in the \(W_s\) vector
    - Subsequent elements \((i > 1)\)
      - Elements up to \(\pi[i - 1]\) are already selected
      - Calculate:
        \[
        \overline{p} = \sum_{j \notin \{\pi[1], \ldots, \pi[i-1]\}} p_{\pi[i-1],j}
        \]
      - If the sum is positive – select \(j \notin \{\pi[1], \ldots, \pi[i - 1]\}\) proportionally to \(\frac{p_{\pi[i-1],j}}{\overline{p}}\)
      - If the sum is zero – select uniformly from \(j \notin \{\pi[1], \ldots, \pi[i - 1]\}\)
EDAs for the FFP

- **The EH-PBIL model**
  - A matrix $\mathbf{P}_{[N \times N]}$ with $p_{i,j} \in [0, 1]^N$
    - $p_{i,j}$ – how probable it is that $j$ is placed right after $i$ in good permutations
  - $\mathbf{P}$ matrix update rule similar to PBIL

- An additional element:
  - A weight vector $\mathbf{W}_s$
    - Weights are used to determine with what probability each element can be used as the first one in the permutation

- Weight vector $\mathbf{W}_s$ update
  - Take the first element $k$ in each solution in the sample $P_{\text{sample}}$
  - Increase $\mathbf{W}_s[k]$ by the evaluation of the solution from which $k$ was taken
Experiments

- Compared algorithms
  - Ant Colony Optimization (ACO)
  - Evolutionary Algorithm (EA) with CX, OBX and PBX crossover and insertion mutation
  - Estimation of Distribution Algorithm with Mallows, Generalized Mallows, EH-PBIL and State-Position models
  - State-Position model with Exponential, Linear, Square and Square root functions
  - Variable Neighbourhood Search (VNS)
Experiments

Test instances

- Graphs: the Erdős-Rényi model $G(N_v, P_{\text{edge}})$
- $N_v = 500, 750, ..., 2250, 2500$ and $5000^*)$
- $P_{\text{edge}} = 3 / N_v$
- Number of starting points: $N_s = 1$
- Number of firefighters per a time step: $N_f = 2$
- The number of edges adjacent to starting points $> N_f$ (to eliminate the possibility of cutting the fire off at $t = 1$)
- Costs assigned to graph vertices drawn from $U[0, 100]$

*) For $N_v = 500, 750, 1000, 2000$ and $5000$ connectedness of the graphs was enforced
Experiments

- Stopping condition
  - Running time
  - $T_{\text{max}} = 300, 600, 900 \text{ and } 7200\text{s}$

- 50 runs

- Median value of the total cost of the saved nodes calculated

- Wilcoxon test applied (null hypothesis: equality of medians)

- Family-Wise Error Rate calculated for comparison of the best performing method vs. the other ones
Parameters – EA

- Population size: $N_{pop} = 100$
- Elitism
- Crossover: CX, OBX, PBX*
- $P_{\text{cross}} = 0.9$
- Insertion mutation *
- $P_{\text{mut}} = 0.05$

Parameters – ACO

- Parameters set according to [1]
  - Population size: $N_{\text{pop}} = 100$
  - Determinism rate:
    - $q_0 = 0.5$
    - with $P = q_0$ the vertex with the highest probability is chosen deterministically
    - with $P = 1 - q_0$ a vertex is chosen randomly proportionally to probabilities
  - Learning rate: $\rho = 0.1$
  - Restart condition: convergence factor > 0.99

Parameters – EDA

- Population size: $N_{\text{pop}} = 100$
- Sample size: $N_{\text{sample}} = 20$
- Elitism

- State-Position Model
  - No tunable parameters
  - Except for the distance-transforming function $f(\cdot)$
Parameters – EDA

- Mallows and Generalized Mallows [1]
  - Initial spread: 1.0
  - Spread limits: [0.00001, 10.0]

- EHBSA [2]
  - $P_{\text{unif}} = 0.4$ and 1.0
  - $B_{\text{ratio}} = 0.0002$


Parameters – EDA

**EH-PBIL**

- Parameters set according to [1]
  - Positive learning rate: $\eta^+ = 0.1$
  - Negative learning rate: $\eta^- = 0.075$
  - Mutation probability: $P_{\text{mut}} = 0.02$
  - Mutation shift parameter: $\mu = 0.05$

- $P_{\text{unif}} = 0.0, 0.2, 0.4, 0.6, 0.8$ and $1.0$

And the winner is...
And the winner is…

- There are two methods that performed best
- For $N_v = 500$ and 750 the VNS
- For $N_v \geq 1250$ the EDA with the State-Position model
- Both results are statistically significant (FWER ≤ 0.0104)
- For $N_v = 1000$ no statistically significant difference - very close results for the VNS ($2.351 \cdot 10^3$) and the S-P EDA ($2.372 \cdot 10^3$)

More details: LNCS 10197 pp. 119-120
Dynamic View

- $N_v = 2500$
- Variants with the best performance for this graph size
Conclusion

- Several classes of algorithms tested on the FFP
- EDAs with various models compared
- An S-P model proposed for representing the relationship between the state of the graph and the defended positions
- The VNS better for smaller graphs
- The S-P EDA better for larger graphs
- Ongoing work
  - EDAs based on subgraph models
  - EDAs for the multiobjective FFP

Thank You! Questions?