Simheuristics for the Multiobjective Nondeterministic Firefighter Problem in a Time-Constrained Setting

Krzysztof Michalak[1)], Joshua D. Knowles[2)]

[1)] Wroclaw University of Economics, Poland [krzysztof.michalak@ue.wroc.pl]
[2)] University of Birmingham, UK [j.knowles.1@cs.bham.ac.uk]

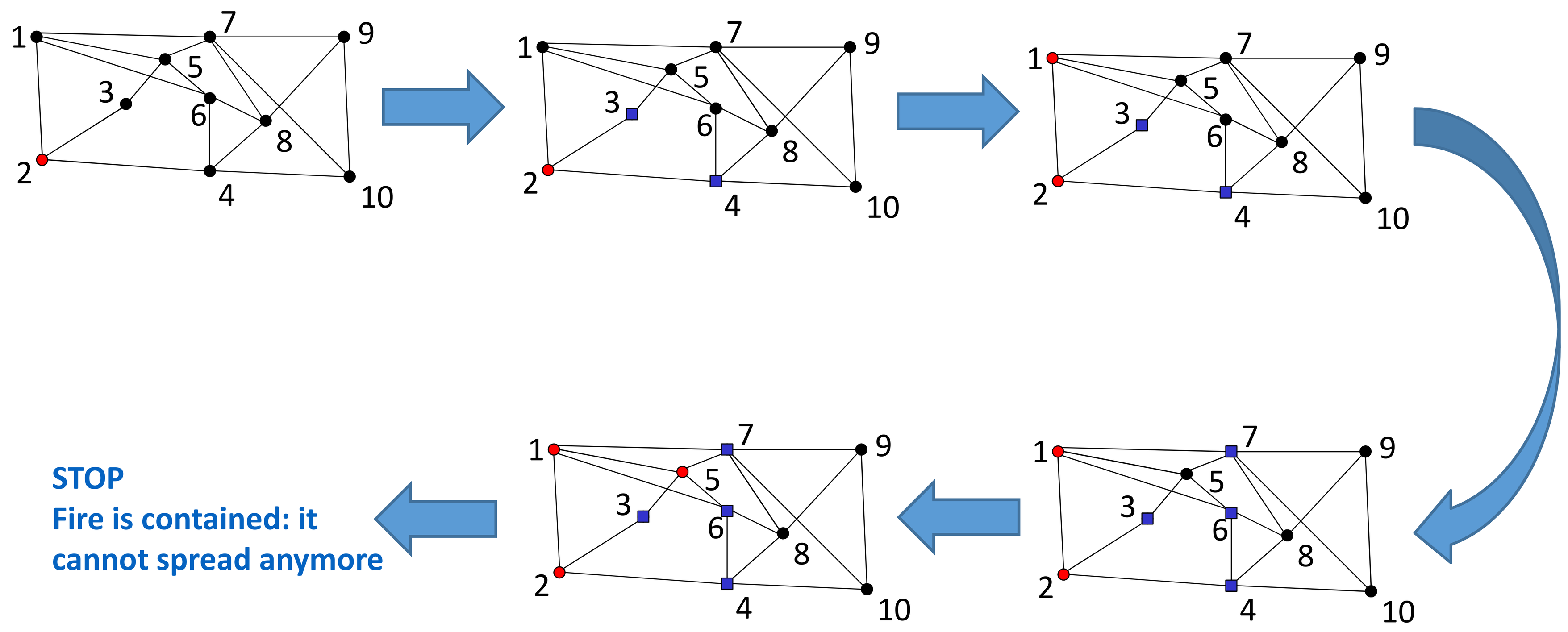
The Firefighter Problem (FFP)

- 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

    let $l(v)$ denote the label assigned to node v
  - Initially a certain number N_s of nodes are burning ('B') and the remaining ones are untouched
- 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

Example ($N_s = 1$, $N_f = 2$)

Solution (permutation): 3, 4, 2, 1, 6, 7, 8, 9, 10, 5



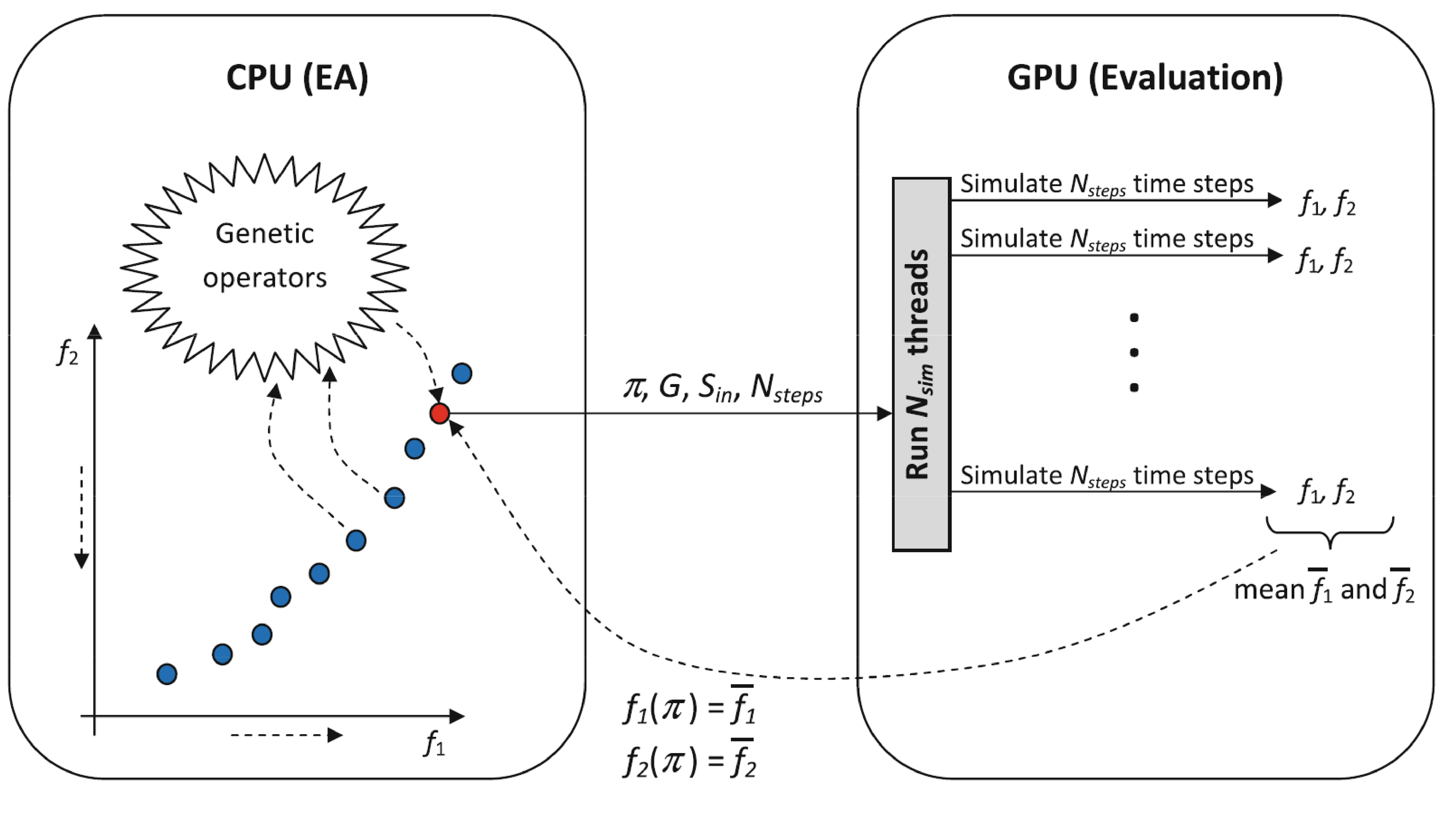
Task: Find the best assignment of firefighters to graph nodes

FFP variants

- Single-objecitve: save the highest possible number of nodes in the graph
- Multi-objective[2]: each node has several values assigned
- Nondeterministic: fire spreads from node to (an adjacent) node with a probability $P_{spread} < 1$
  - Dynamic Optimization Problem (DOP)
  - Possible approaches: offline and online optimization

Simheuristics[3]

- The main algorithm is a heuristic (e.g. EA)
- Spread of fire is simulated to evaluate solutions
- We perform $N_{sim} = 200$ simulations and average the results
- CPU+GPU architecture is used



Implementation of the simheuristic on a CPU/GPU machine.

The Sim-EA Algotihm

The original idea – solve several similar instances of a problem at the same time

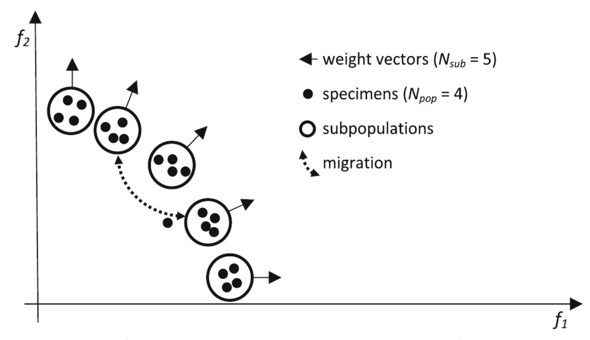
- A multipopulation algorithm…
- …but each subpopulation tackles a slightly different instance of the problem
- Island model + migration
- Migration influenced by similarities of problem instances

The Sim-EA for Multiobjective Optimization

- Decomposition-based approach
- Each subpopulation tackles a subproblem with a different weight vector
- Problem similarity = a dot product of their weight vectors

The Sim-EA for the Multiobjective FFP

- A nondeterministic version of the problem
- Dynamic optimization
- Online interaction with the developing fire
- We need to simulate each trade-off between the objectives separately



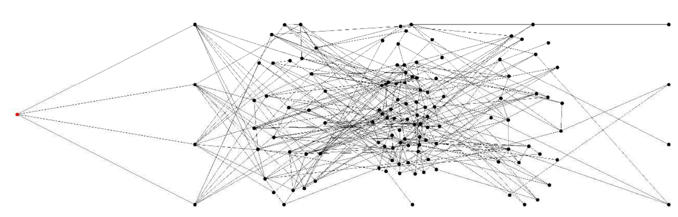
An overview of elements of the Sim-EA algorithm.

Experiments and Results

Test data

Graphs with N_v = 30, 40, 50, 75, 100 and 125

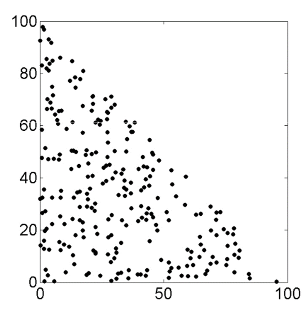
For each pair of vertices an edge is added with a probability $P_{edge} = 2.5 / N_v$



$N_s = 1$, $N_f = 2$

Cost assignment

- Pairs of random values with a uniform probability on a triangle formed by points [0, 0], [0, 100], [100, 0]
- Sum of costs associated with a vertex cannot exceed 100
- It is not possible to maximize both objectives at the same time



Comparing the online and offline optimization

- For smaller instances (N ≤ 50) both optimization approaches are more or less comparable
- For larger instances (N ≥ 75) the online optimizer outperformed the offline one 12 times (with 3 cases undecided, none in favour of the offline optimizer)

The results of a statistical comparison of the online and offline optimization modes: 'N' - online better, 'F' - offline better, '=' - no statistical difference.

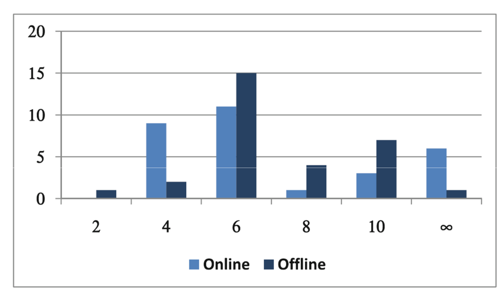
N_v	$P_{sp} = 0.3$	$P_{sp} = 0.5$	$P_{sp} = 0.7$	$P_{sp} = 0.9$	$P_{sp} = 1.0$
30	=	N	N	=	=
40	N	=	=	=	=
50	=	N	=	=	F
75	=	N	N	N	=
100	N	N	N	N	=
125	N	N	N	N	N

The influence of the number of simulation steps N_{step}

Letting the simulation run without limit can deteriorate the results

For larger instances (N_v = 75, 100 and 125):

- The unlimited simulation length worked best for $P_{sp} \le 0.5$
- For $P_{sp} \ge 0.7$ limiting the duration of simulations worked better
- See Tables 1-6 in the paper



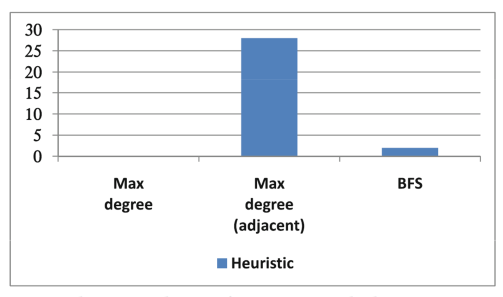
The number of times each value of N_{steps} produced the best result.

Optimization vs.heuristics

Best: select the vertex with highest degree adjacent to the fire

Comparison with the optimizers:

- $P_{sp} \le 0.5$ – results in favour of the heuristics
- $P_{sp} \ge 0.7$ – results in favour of the optimizers



The number of times each heuristic produced the best result.

Hypothesis

For low P_{sp} it is hard to obtain good evaluations of solutions using simulations

- Unlimited simulation length worked best for $P_{sp} \le 0.5$
- The "Max degree (adjacent)" heuristic is very effective when $P_{sp} \le 0.5$

References

[1] Hartnell, B.: *„Firefighter! An application of domination"*, in: 20th Conference on Numerical Mathematics and Computing (1995).

[2] K. Michalak *„Auto-adaptation of Genetic Operators for Multi-objective Optimization in the Firefighter Problem"*, Intelligent Data Engineering and Automated Learning IDEAL 2014, Lecture Notes in Computer Science, vol. 8669, pp. 484-491, Springer, 2014.

[3] Juan, A.A.: *"A review of simheuristics: extending metaheuristics to deal with stochastic combinatorial optimization problems"*. Oper. Res. Perspect. 2, 62–72 (2015).

[4] K. Michalak *„The Sim-EA Algorithm with Operator Autoadaptation for the Multiobjective Firefighter Problem"*, 15th European Conference, EvoCOP 2015, Copenhagen, Denmark, April 8-10, 2015, Proceedings, Lecture Notes in Computer Science, volume 9026, str. 184–196, Springer, 2015.