Solving the Firefighter Problem with Two Elements using a Multi-modal Estimation of Distribution Algorithm

Piotr Lipinski*
* Computational Intelligence Research Group
Institute of Computer Science
University of Wroclaw, Wroclaw, Poland
lipinski@cs.uni.wroc.pl

***Abstract*—The Firefighter Problem (FFP) is an optimization problem of developing an optimal strategy for assigning firemen to nodes of a given graph in successive iterations of a simulation of spread of fires in the graph. This paper focusses on an extension of the original FFP, namely the Bi-Firefighter Problem (FFP2), where the second element (water) is introduced. FFP2 corresponds to the practical optimization problems, where more than one disease is spreading in the environment, and the objective is to minimize the total loss. Since the loss may come from two different sources, each of which causes different damages, the objective function is more complex than in the case of the original FFP. In this paper, an evolutionary approach to FFP2, the EA-FFP2 algorithm, based on a multi-modal Estimation of Distribution Algorithm (EDA), is proposed. EA-FFP2 was validated on a number of benchmark FFP2 instances that were also solved by the branch and bound algorithms or the heuristic local search algorithms run for a long time. Computational experiments confirmed that EA-FFP2 was capable of solving FFP2 and finding solutions close to the optima determined by the branch and bound algorithms or to the quasi-optima determined by exhaustive local search.**

I. Introduction

The Firefighter Problem (FFP), introduced by Bert Hartnell in 1995 [14], is an optimization problem of developing an optimal strategy for assigning firemen to nodes of a given graph in successive iterations of a simulation of spread of fires in the graph. In 2007, Finbow et al. proved that FFP is NP-complete for trees of maximum degree 3, but it may be solved in a polynomial time for graphs of maximum degree 3 if fires break out at a node of degree at most 2 [11].

Although FFP is formulated in terms of spread of fires, similar objectives refer also to the spread of epidemic diseases on connected cities, the spread of hazards in electric networks or water systems, the spread of malware in computer networks, as well as, to social media mining, where the graph represents the social network and the spread of fires represents the information flow, especially the spread of marketing information in the social network. Similar problems are also related to financial data analysis, where the graph represents dependencies between financial instruments and the spread of fires represents the capital flow.

Some theoretical approaches try to distinguish the properties of FFP that enable to solve it efficiently. Some studies focus on the structure of the graph: Three and more dimensional grids are considered in [8]. Planar graphs are discussed in [9]. Infinite graphs are studied in [21]. A number of firefighters is another key factor for the problem complexity: [3] studies FFP with more than one firefighter on trees. [7] considers FFP as a game, discusses properties of optimal strategies for trees, and proposes efficient approximation algorithms. [10] analyzes how many firefighters per turn are needed to stop the fire and estimates the lower and upper bounds for planar grids. Some results refer to the surviving rate of a graph in FFP: In [6], approximations for the surviving rates of trees and outerplanar graphs are proposed. In [5], the surviving rates of graphs with bounded treewidth are discussed. A number of approaches concerns also approximation algorithms for FFP [2], [1], [16]. A survey on FFP may be found in [12].

Recently, a number of computational approaches that try to solve FFP using heuristic optimization methods has appeared. In [15], the Variable Neighborhood Search (VNS) algorithm with a new representation of the solution to FFP was proposed. In [13], integer linear programming and heuristic algorithms were used to solve FFP with random graphs. In [4], a metaheuristic approach based on the Ant Colony Optimization (ACO) algorithm was proposed.

Beside the original FFP, there are some popular extensions of the original optimization problem. The multi-objective FFP was introduced in [22], and an evolutionary algorithm with auto-adaptation and a number of crossover operators was proposed to solve it. In [23], a multipopulation evolutionary algorithm for the multi-objective FFP was discussed. The non-deterministic FFP was introduced in [24], and a multipopulation evolutionary algorithm was proposed to solve it.

This paper focusses on an extension of the original FFP, namely the Bi-Firefighter Problem (FFP2), where the second element (water) is introduced. FFP2 corresponds to the practical optimization problems, where more than one disease is spreading in the environment, and the objective is to minimize the total loss. Since the loss may come from two different sources, each of which causes different damages, the objective

978-1-5090-4601-0/17/$31.00

function is more complex than in the case of the original FFP.

In this paper, an evolutionary approach to FFP2, based on a multi-modal Estimation of Distribution Algorithm (EDA) [17], is proposed. It aims at constructing a probability distribution that is capable of generating efficient candidate solutions to the optimization problem. It starts with an initial probability model and tries to improve it in successive iterations by sampling a random population, evaluating the candidate solutions, and estimating a new probability model from the promising candidate solutions.

This paper is structured in the following manner: Section II defines FFP and FFP2. Section III describes the baseline heuristic local search approach. Section IV describes the proposed evolutionary approach. Section V presents the results of the experiments. Finally, Section VI concludes the paper.

II. Problem Definition

The Firefighter Problem (FFP) is an optimization problem defined in the context of a certain undirected graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{N_V}\}$ is the set of nodes ($N_V = |\mathbf{V}|$ denotes the number of nodes of the graph $\mathcal{G}$), and $\mathbf{E} \subset \{\{\mathbf{v}_i, \mathbf{v}_j\} : \mathbf{v}_i, \mathbf{v}_j \in \mathbf{V} \text{ and } \mathbf{v}_i \neq \mathbf{v}_j\}$ is the set of edges. Such a graph defines a topographic structure, where fires had appeared and spread throughout the structure in successive time instants. In each time instant t, each node $\mathbf{v}_i \in \mathbf{V}$ may be in one of three possible states, 'untouched' (U), 'burning' (B), 'defended' (D), denoted by $s_i^{(t)} \in \{\mathrm{U}, \mathrm{B}, \mathrm{D}\}$, so the state of the entire structure may be described by the vector $\mathbf{s}^{(t)} = (s_1^{(t)}, s_2^{(t)}, \ldots, s_{N_V}^{(t)}) \in \{\mathrm{U}, \mathrm{B}, \mathrm{D}\}^{N_V}$. The initial state of the structure $\mathbf{s}^{(0)}$ in time $t = 0$ is given, and it changes in successive time instants in the following manner: In each time instant t, a certain number N_F of firemen is located at untouched nodes changing them to defended nodes, and then fires spread to adjacent not defended nodes. The simulation continues until the fire is stopped, i.e. there are no more nodes that change their state. Formally: For the number N_F of selected nodes $\mathbf{v}_i$ such that $s_i^{(t-1)} = \mathrm{U}$, $s_i^{(t)} = \mathrm{D}$ (the selection of that nodes is an element of the strategy being the solution to FFP). For the other nodes $\mathbf{v}_i$, if $s_i^{(t-1)} = \mathrm{U}$ and there is an edge $\{\mathbf{v}_i, \mathbf{v}_j\} \in \mathbf{E}$ and a node $\mathbf{v}_j \in \mathbf{V}$ such that $s_j^{(t-1)} = \mathrm{B}$, $s_i^{(t)} = \mathrm{B}$, otherwise $s_i^{(t)} = s_i^{(t-1)}$. A solution to FFP is a strategy of selecting nodes to defend in successive time instants. It may be represented by a permutation $\mathbf{p}$ of nodes $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{N_V}$, in such a way that, in each time instant t, the first N_V nodes of the permutation that were still untouched in time instant $t-1$ are taken. Different permutations may lead to different simulations and different final states. FFP consists in finding a permutation that maximizes the number of untouched or defended nodes in the final state.

One of popular extensions of FFP introduces weights assigned to nodes, representing the losses related to the fire, and defines the objective function to minimize as the sum of weights of the burning nodes in the final state. It may be reformulated to the maximization problem with the objective function defined as the sum of weights of the untouched and defended nodes.

Figure 1 (a) - (c) illustrates the definition of FFP on a graph of 10 nodes. Figure 1 (a) presents the initial state: the node 5 is burning, the other nodes are untouched. Figure 1 (b) presents the next state: 2 firemen are located at the nodes 6 and 8, fires spread to the nodes 7 and 9. Figure 1 (c) presents the third state: 2 other firemen are located at the nodes 1 and 2, fires spread to the node 3. It is the final state, because fires cannot spread to other nodes. Firemen were located according to a permutation $(8, 5, 6, 7, 1, 2, 4, 0, 9, 3)$ (but the same result may be obtained with some other permutations, for instance $(5, 8, 6, 9, 7, 2, 1, 4, 3, 0)$). The value of the objective function for the permutation equals 6 (2 nodes were untouched and 4 nodes were defended).

This paper concerns the Bi-Firefighter Problem (FFP2) being an extension of FFP with a second element (water). Beside fire, flood is considered, so each node $\mathbf{v}_i$ may be in one of five possible states, 'untouched' (U), 'burning' (B), 'defended' (D), 'flooded' (F), 'burning and flooded' (BF), denoted by $s_i^{(t)} \in \{\mathrm{U}, \mathrm{B}, \mathrm{D}, \mathrm{F}, \mathrm{BF}\}$. The simulation is similar to FFP, but with flood that also spreads. Each node has two weights, w_i and u_i, assigned: the first represents losses related to fire, the second represents losses related to flood. The objective function to minimize is the sum of weights w_i of the burning or burning and flooded nodes plus the sum of weights u_i of the flooded or burning and flooded nodes in the final state. Different losses related to fire and to flood, especially cumulated in burning and flooded nodes, make the optimization problem more complex and difficult to solve by the approaches designed for the original FFP.

Figure 1 (d) - (f) illustrates the definition of FFP2 on a graph of 10 nodes. Figure 1 (d) presents the initial state: the node 5 is burning, the node 3 is flooded, the other nodes are untouched. Figure 1 (e) presents the next state: 2 firemen are located at the nodes 6 and 8, fires spread to the nodes 7 and 9, flood spread to the nodes 2 and 9 (the node 9 is burning and flooded). Figure 1 (f) presents the third state: 2 other firemen are located at the nodes 1 and 4, fires spread to the node 3, flood spread to the node 0, 5, 7. Firemen were located according to a permutation $(8, 5, 6, 7, 1, 2, 4, 0, 9, 3)$ (but the same result may be obtained with some other permutations). The value of the objective function for the permutation equals $w_3 + w_5 + w_7 + w_9 + u_0 + u_2 + u_3 + u_5 + u_7 + u_9$.

III. A Heuristics Local Search Approach

Solutions to the FFP and FFP2 instances with relatively small graphs may be found by full search methods or branch and bound methods. For the FFP and FFP2 instances with N_V in the order of hundreds, as the search space consists of $N_V!$ candidate solutions, even branch and bound methods may run too long (especially, if the optimization problem is considered with the weights assigned to nodes, which introduces additional difficulties for branching and bounding, because two different candidate solutions leading to the same numbers of burning and flooded nodes may differ in terms of

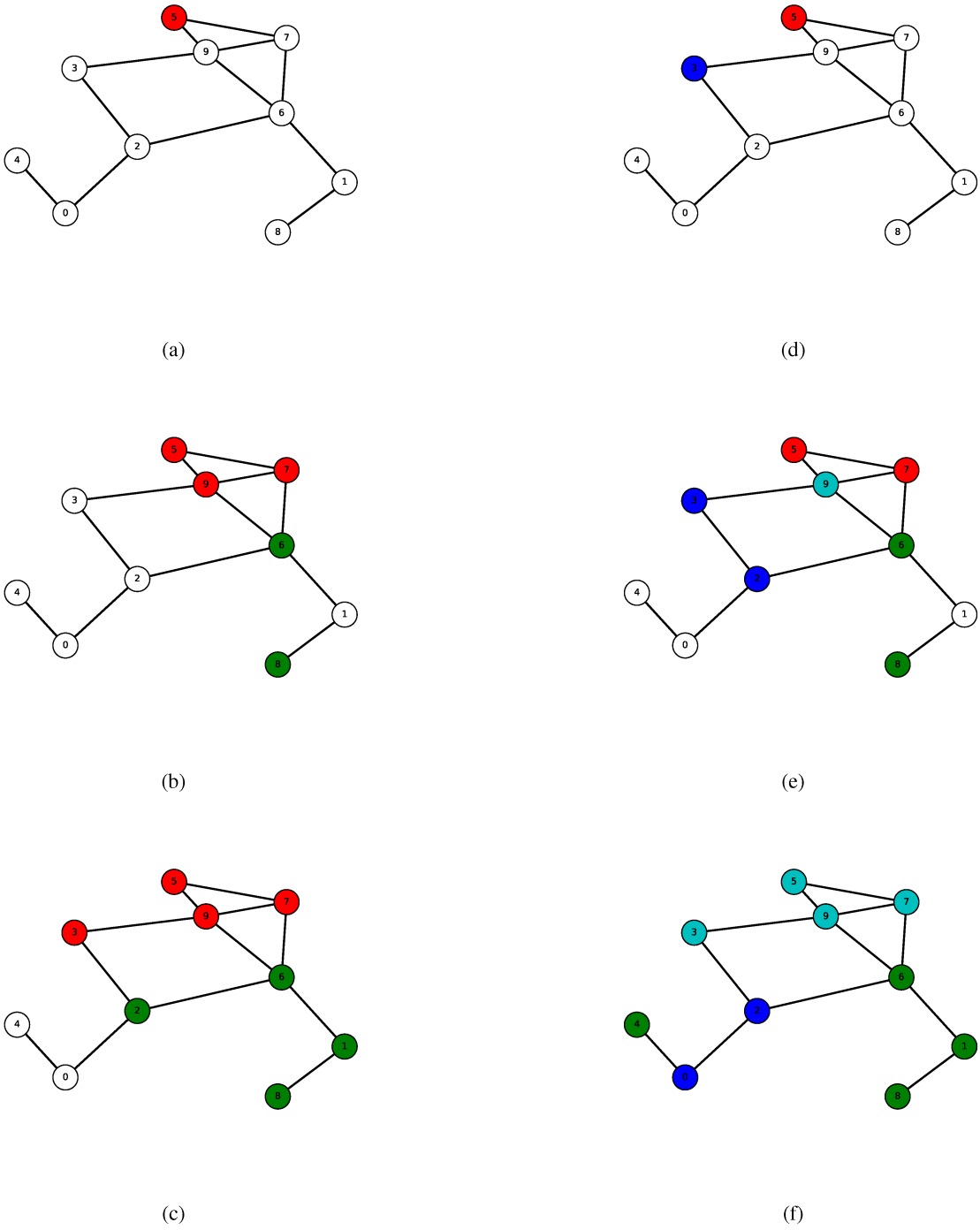

Fig. 1: Illustration of the definition of FFP and FFP2 on a graph of 10 nodes: (a) presents the initial state in FFP, (b) presents the next state, and (c) presents the third state, (d), (e), (f) presents the same for FFP2. Burning nodes are red, defended nodes are green, flooded nodes are blue, burning and flooded nodes are cyan, and untouched nodes are white.

the objective function). However, similarly to many other complex optimization problems [18], the baseline quasi-optimal solutions may be evaluated by simple heuristic local search algorithms run for a large number of iterations for a long time [19], [20].

A few approaches to determine the baseline quasi-optimal solutions were examined. Each of them starts with a random candidate solution generated as a random permutation with

the uniform probability distribution over Π_{N_V} (Π_n denotes the set of all permutations of length n) and tries to improve the candidate solution using local search methods. Due to the size constraint, only two approaches, GLS and ILS-SA, are described in this paper.

The first approach is based on Greedy Local Search (GLS), presented in Algorithm 1, which aims at transforming the input candidate solution $\mathbf{x}_0$ into the nearest local optimum $\mathbf{x}$. Starting with the candidate solution $\mathbf{x} = \mathbf{x}_0$, the neighborhood of $\mathbf{x}$ is scanned for the best candidate solution $\hat{\mathbf{x}}$ (the neighborhood of a permutation $\mathbf{x}$ is defined by a set of all possible permutations $\mathbf{y}$ that may be transformed to $\mathbf{x}$ by no more than ϱ transpositions, where ϱ is a constant parameter denoting the radius of the neighborhood). If $\hat{\mathbf{x}}$ outperforms $\mathbf{x}$, it replaces $\mathbf{x}$, and the above process repeats until there are no more improvement.

Algorithm 1 Greedy Local Search (GLS)

Require: $\mathbf{x_0}$: the initial candidate solution
Ensure: $\mathbf{x}$: the best found candidate solution
  $\mathbf{x} = \mathbf{x}_0$
  $\hat{\mathbf{x}}$ = Best-Solution-In-Neighborhood($\mathbf{x}$)
  while $F(\hat{\mathbf{x}}) < F(\mathbf{x})$ **do**
    $\mathbf{x}$ = $\hat{\mathbf{x}}$;
    $\hat{\mathbf{x}}$ = Best-Solution-In-Neighborhood($\mathbf{x}$)
  end while

However, scanning over the entire neighborhood of $\mathbf{x}$ requires evaluating $\mathcal{O}(N_V^{\varrho})$ candidate solutions, which, multiplied by the number of iterations of GLS, leads to a very long computing time for large N_V, even for $\varrho = 5$. Therefore, instead of reducing ϱ, the scanning is replaced with the random partial scanning that evaluates only a given number of randomly chosen candidate solutions from the neighborhood (in the experiments, N_V randomly chosen candidate solutions).

Moreover, a single run of GLS stops in the nearest to the input candidate solution $\mathbf{x}_0$ local optimum $\mathbf{x}$. Since, in complex optimization problems, there are usually many local optima, a single run of GLS is not capable of reaching the global optimum, and it should be restarted many times.

The second approach is based on Iterative Local Search - Simulated Annealing (ILS-SA), presented in Algorithm 2, which tries to run GLS multiple times with different starting candidate solutions, similarly to Simulated Annealing. It starts with initializing the best found candidate solution $\mathbf{x}$ and the base candidate solution $\bar{\mathbf{x}}$ by the input candidate solution $\mathbf{x}_0$ improved by GLS. In successive iterations, the base candidate solution $\bar{\mathbf{x}}$ is disturbed by τ random transpositions, where τ is a constant parameter (in the experiments, $\tau = 5$), and improved by GLS, giving the candidate solution $\hat{\mathbf{x}}$. If $\hat{\mathbf{x}}$ outperforms $\mathbf{x}$, it replaces $\mathbf{x}$ and $\bar{\mathbf{x}}$. Otherwise, with a small probability decreasing in successive iterations, $\hat{\mathbf{x}}$ replaces $\bar{\mathbf{x}}$.

Table I presents a comparison of the results of ILS-SA with the exact optimal solutions found using the branch and bound approach for 20 benchmark FFP2 instances with graphs of $N_V = 25$ nodes. Each row refers to one FFP2 instance

Algorithm 2 Iterative Local Search - Simulated Annealing (ILS-SA)

Require: $\mathbf{x_0}$: the initial candidate solution
Ensure: $\mathbf{x}$: the best found candidate solution
  $\mathbf{x}$ = Greedy-Local-Search($\mathbf{x}_0$)
  $\bar{\mathbf{x}}$ = $\mathbf{x}$
  $t = 1$
  while $t < T$ **do**
    $\tilde{\mathbf{x}}$ = Mutation($\bar{\mathbf{x}}$, τ)
    $\hat{\mathbf{x}}$ = Greedy-Local-Search($\tilde{\mathbf{x}}$)
    if $F(\hat{\mathbf{x}}) < F(\mathbf{x})$ **then**
      $\mathbf{x}$ = $\hat{\mathbf{x}}$
      $\bar{\mathbf{x}}$ = $\hat{\mathbf{x}}$
    else
      if random() $< \exp(-(F(\hat{\mathbf{x}}) - F(\mathbf{x})) \cdot t/T)$ **then**
        $\bar{\mathbf{x}}$ = $\hat{\mathbf{x}}$
      end if
    end if
    $t = t + 1$;
  end while

(described in Section V) with $N_V = 25$ nodes, N_b burning nodes and $N_f = N_b$ flooded nodes in the initial state, and the number of firemen $N_F = N_b = N_f$. The third column contains the optimum found by the branch and bound approach. For each problem instance, ILS-SA was run 100 times for $T = 500$ iterations (but it was stopped, if the computing time exceeded 48 hours, which occurred only few times) with the number of transpositions $\tau = 5$. The number of GLS iterations was limited to 10 (in most cases GLS stopped earlier, after a few iterations, but the constraint was added to avoid some inefficient computation in some cases), and the random partial scanning with N_V randomly chosen candidate solutions was used in GLS. The next four columns present the minimum, maximum, mean and standard deviation for the 100 runs of ILS-SA. The last two columns contain the quasi-optimum found by ILS-SA and the difference between it and the exact optimum. Clearly, ILS-SA run for a large number of iterations for a long time was capable of finding quasi-optimal solutions very close to exact optima. In order to evaluate the statistical significance, the Mann Whitney Wilcoxon test was performed. The p-value equal to 0.9676 proved that there were no significant differences between the results of ILS-SA and the results of the branch and bound approach. In addition, the Wilcoxon test was also performed, and the p-value equal to 0.1970 confirmed the previous conclusions.

Therefore, in the experiments reported in this paper, ILS-SA was used to determine the baseline quasi-optimal solutions to the benchmark FFP2 instances with larger numbers of nodes.

IV. An Evolutionary Approach

Although the heuristic local search approach is capable of finding quasi-optimal solutions very close to exact optima, it requires large numbers of iterations and long computing times. An evolutionary approach, proposed in this paper, tries to solve

TABLE I: A comparison of ILS-SA with the branch and bound approach on 20 benchmark FFP2 instances with graphs of 25 nodes. ILS-SA was run 100 times for each problem instance

			ILS-SA					
		optimum	min	max	mean	std	quasi-optimum	error
1	$N_b = N_f = N_F = 1$	553.7438	553.7438	565.0177	554.1947	2.2092	553.7438	0.0000
2		339.6817	339.6817	611.0160	383.9325	99.1537	339.6817	0.0000
3		265.5675	265.5675	265.5675	265.5675	0.0000	265.5675	0.0000
4		234.8902	234.8902	234.8902	234.8902	0.0000	234.8902	0.0000
5		257.6267	257.6267	257.6267	257.6267	0.0000	257.6267	0.0000
6		270.2774	270.2774	270.2774	270.2774	0.0000	270.2774	0.0000
7		158.7421	158.7421	159.1323	158.7460	0.0388	158.7421	0.0000
8		146.1840	146.1840	148.1735	146.3631	0.5693	146.1840	0.0000
9		227.1510	227.1510	227.1510	227.1510	0.0000	227.1510	0.0000
10		47.6519	47.6519	47.6519	47.6519	0.0000	47.6519	0.0000
11	$N_b = N_f = N_F = 2$	541.6537	541.6537	541.6537	541.6537	0.0000	541.6537	0.0000
12		332.3805	332.3805	427.0635	356.8753	38.9673	332.3805	0.0000
13		289.5725	289.5725	483.1967	302.3654	46.0467	289.5725	0.0000
14		193.8031	193.8031	193.8031	193.8031	0.0000	193.8031	0.0000
15		127.8106	127.8106	127.8106	127.8106	0.0000	127.8106	0.0000
16		216.5017	216.5017	216.5017	216.5017	0.0000	216.5017	0.0000
17		257.9487	257.9487	257.9487	257.9487	0.0000	257.9487	0.0000
18		505.4304	510.5493	510.5493	510.5493	0.0000	510.5493	5.1189
19		198.0368	198.0368	198.0368	198.0368	0.0000	198.0368	0.0000
20		203.0432	203.0432	203.0432	203.0432	0.0000	203.0432	0.0000

the optimization problem faster with probability modeling instead of exhaustive local search.

The evolutionary approach is based on a multi-modal Estimation of Distribution Algorithm (EDA) [17], which aims at constructing a probability distribution that is capable of generating efficient candidate solutions to the optimization problem. It starts with an initial probability model and tries to improve it in successive iterations by sampling a random population, evaluating the candidate solutions, and estimating a new probability model from the promising candidate solutions. Algorithm 3 presents the overview of the approach.

Algorithm 3 Evolutionary Algorithm for FFP2 (EA-FFP2)

$\mathbb{M} =$ Initial-Model()
while not Termination-Condition($\mathbb{M}$) **do**
  $\mathcal{P} =$ Random-Population($\mathbb{M}, N$)
  Population-Evaluation($\mathcal{P}, F$)
  $\mathbb{M} =$ Model-From-Population($\mathcal{P}$)
end while

The probability model $\mathbb{M}$ used in the proposed algorithm is a mixture of K probability distributions $\mathbb{P}_1, \mathbb{P}_2, \ldots, \mathbb{P}_K$ with probabilities $p_1, p_2, \ldots, p_K$. Each probability distribution $\mathbb{P}_k$ is defined by the mean $\mathbf{p}_k \in \Pi_{N_V}$ and the range $\tau_k \in \{1, 2, \ldots\}$. Generating a random candidate solution $\mathbf{x}$ with the model $\mathbb{M}$ has two steps: It begins with random choosing the number $k \in \{1, 2, \ldots, K\}$ of the probability distribution to use (with the probability that $k = i$ equal to p_i). After that, the candidate solution $\mathbf{x}$ is generated using the k-th probability distribution $\mathbb{P}_k$ in such a way that $\mathbf{x}$ is the permutation $\mathbf{p}_k$ transformed τ_k times by a random transposition.

At the beginning, the model $\mathbb{M}$ is initalized with the means $\mathbf{p}_k$ being random permutations improved by GLS, the ranges τ_k being random integers between 1 and τ_0, where τ_0 is a constant parameter (in the experiments, $\tau_0 = 4$), and equal probabilities $p_k = 1/K$. Algorithm 4 presents the model initialization process.

Algorithm 4 Initial-Model

for $k = 1, 2, \ldots, K$ **do**
  $\mathbf{p}_k =$ Random-Permutation(N_V)
  $\mathbf{p}_k =$ Greedy-Local-Search($\mathbf{p}_k$)
  $\tau_k =$ Random-Integer(τ_0)
  $\mathbb{P}_k = (\mathbf{p}_k, \tau_k)$
  $p_k = 1/K$
end for
$\mathbb{M} = ((\mathbb{P}_1, \mathbb{P}_2, \ldots, \mathbb{P}_K), (p_1, p_2, \ldots, p_K))$

In successive iterations, the model $\mathbb{M}$ is updated on the basis of the current population $\mathcal{P}$. First, each candidate solution $\mathbf{x} \in \mathcal{P}$ is improved by GLS, and K best unique candidate solutions are chosen, denoted by $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K$. Second, for each $k = 1, 2, \ldots, K$, a set of all candidate solutions $\mathbf{x} \in \mathcal{P}$ generated with the same probability distribution as $\mathbf{x}_k$ is determined, denoted by $\mathbf{S}_k$ (if q denotes the index of the probability distribution used to generate $\mathbf{x}_k$, then $\mathbf{S}_k = \{\mathbf{x} \in \mathcal{P} : \mathbf{x}$ was generated using $\mathbb{P}_q\}$). Clearly, $\mathbf{S}_k$ is never empty, because $\mathbf{x}_k \in \mathbf{S}_k$. Next, a kind of a diameter diam($\mathbf{S}_k$) of $\mathbf{S}_k$ is evaluated as an average of Kendall Tau distances[1] over all pairs $(\mathbf{x}, \mathbf{y})$ of permutations from $\mathbf{S}_k$, where

[1]Kendall Tau distance for a pair of permutations $(\mathbf{x}, \mathbf{y})$ is $\tau(\mathbf{x}, \mathbf{y}) = (C - D)/(C + D)$, where C is the number of concordant pairs of positions, i.e. $C = |\{(i, j) : (x_i < x_j$ and $y_i < y_j)$ or $(x_i > x_j$ and $y_i > y_j), i, j = 1, 2, \ldots, d, i \neq j\}|$, and D is the number of discordant pairs of positions, i.e. $D = |\{(i, j) : (x_i < x_j$ and $y_i > y_j)$ or $(x_i > x_j$ and $y_i < y_j), i, j = 1, 2, \ldots, d, i \neq j\}|$, where d denotes the length of the permutations.

$\mathbf{x} \neq \mathbf{y}$ (if $\mathbf{S}_k$ consists only of $\mathbf{x}_k$, $\text{diam}(\mathbf{S}_k) = 0.5$). Finally, the model is redefined with the means $\mathbf{p}_k = \mathbf{x}_k$, the ranges $\tau_k = \max(\tau_0, \tau_1 \cdot \text{diam}(\mathbf{S}_k))$, where τ_1 is a constant parameter (in the experiments, $\tau_1 = 8$), and the probabilities p_k equal to

$$p_k = \frac{\sum_{\mathbf{x} \in \mathbf{S}_k} F(\mathbf{x})}{\sum_{\mathbf{x} \in \mathcal{P}} F(\mathbf{x})},$$

which measures how significant the k-th probability distribution is in terms of improving the objective function. Algorithm 5 presents the model updating process.

Algorithm 5 Model-From-Population

for $\mathbf{x} \in \mathcal{P}$ **do**
  $\mathbf{x} = \text{Greedy-Local-Search}(\mathbf{x})$
end for
$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_K = K$ best unique candidate solutions from $\mathcal{P}$
for $k = 1, 2, \ldots, K$ **do**
  $\mathbf{S}_k$ = set of all candidate solutions from $\mathcal{P}$ generated with the same probability $\mathbb{P}_q$ as $\mathbf{x}_k$
  $\text{diam}(\mathbf{S}_k)$ = average of Kendall Tau distances over all pairs of permutations from $\mathbf{S}_k$
  $\mathbf{p}_k = \mathbf{x}_k$
  $\tau_k = \max(\tau_0, \tau_1 \cdot \text{diam}(\mathbf{S}_k))$
  $\mathbb{P}_k = (\mathbf{p}_k, \tau_k)$
  $p_k = \frac{\sum_{\mathbf{x} \in \mathbf{S}_k} F(\mathbf{x})}{\sum_{\mathbf{x} \in \mathcal{P}} F(\mathbf{x})}$
end for
$\mathbb{M} = ((\mathbb{P}_1, \mathbb{P}_2, \ldots, \mathbb{P}_K), (p_1, p_2, \ldots, p_K))$

V. Experiments

A. The Benchmark Problems

Experiments were performed on a set of benchmark FFP2 instances, created with various graphs $\mathcal{G}$, various numbers of nodes N_V, various vectors of weights $\mathbf{w}$ and $\mathbf{u}$, various initial states $\mathbf{s}^{(0)}$ and various numbers of firemen N_F. Table II lists the problem instances used in the experiments.

Each problem instance concerns a graph $\mathcal{G}$ of a given number N_V of nodes with randomly generated edges (each possible edge was considered and it was included in the graph with the probability $2.5/N_V$), randomly generated vectors of weights $\mathbf{w}$ and $\mathbf{u}$ with the uniform distribution over $[0, 1]^{N_V}$ and then normalized to sum up to 500 each, an initial state $\mathbf{s}^{(0)}$ with a given number N_b of random burning and a given number N_f of random flooded nodes generated with the uniform distribution, and a given number of firemen N_F.

Benchmark FFP2 instances were solved by the branch and bound approach (for $N_V = 25$ only, Table I presents the results) and by ILS-SA to determine the baseline quasi-optimal solutions. For each problem instance, ILS-SA was run 100 times for $T = 500$ iterations (but it was stopped, if the computing time exceeded 48 hours, which occurred only few times) with the number of transpositions $\tau = 5$. The number of GLS iterations was limited to 10 (in most cases GLS stopped earlier, after a few iterations, but the constraint was added to avoid some inefficient computation in some cases), and the random partial scanning with N_V randomly chosen candidate solutions was used in GLS.

TABLE II: List of FFP2 instances used in the experiments

$N_V = 25$	10 instances with $N_b = 1$, $N_f = 1$, $N_F = 1$ 10 instances with $N_b = 2$, $N_f = 2$, $N_F = 2$
$N_V = 50$	10 instances for each $N_b = 1, 2, \ldots, 5$ $N_f = N_b$, $N_F = \max(N_b, 5)$ ($10 \cdot 5 = 50$ instances in total)
$N_V = 100$	10 instances for each $N_b = 1, 2, \ldots, 10$ $N_f = N_b$, $N_F = \max(N_b, 5)$ ($10 \cdot 10 = 100$ instances in total)
$N_V = 150$	10 instances for each $N_b = 1, 2, \ldots, 15$ $N_f = N_b$, $N_F = \max(N_b, 5)$ ($10 \cdot 15 = 150$ instances in total)
$N_V = 200$	10 instances for each $N_b = 1, 2, \ldots, 20$ $N_f = N_b$, $N_F = \max(N_b, 5)$ ($10 \cdot 20 = 200$ instances in total)

B. Results of the Experiments

In order to validate the proposed evolutionary algorithm, EA-FFP2 was run for each benchmark FFP2 instance 10 times for $T = 100$ iterations, with the mixture of $K = 5$ probability distributions, the population composed of 50 candidate solutions, and the parameters $\tau_0 = 4$ and $\tau_1 = 8$. The number of GLS iterations was limited to 10, and the random partial scanning with N_V randomly chosen candidate solutions was used in GLS. Table III presents the results of EA-FFP2 on 50 benchmark FFP2 instances with graphs of $N_V = 50$ nodes. Each row refers to one FFP2 instance. The third column contains the quasi-optimum found by ILS-SA. The next four columns present the minimum, maximum, mean and standard deviation for the 10 runs of EA-FFP2. The next columns report the MAE and MAPE errors (the mean taken from 10 runs of EA-FFP2). MAE is the mean absolute error evaluated as the mean of the difference between the quasi-optimum and the best solution found by EA-FFP2 in each run. MAPE is the mean absolute percentage error evaluated as the mean of the difference between the quasi-optimum and the best solution found by EA-FFP2 in each run divided by the quasi-optimum. The last column contains the number of runs when EA-FFP2 succeeded to find the exact quasi-optimum.

It is easy to see that EA-FFP2 was capable of finding solutions close to quasi-optima determined by exhaustive local search in long computing time. For many benchmark FFP2 instances, even the exact quasi-optimum was reached.

Table IV presents the summary of results of EA-FFP2 on the benchmark FFP2 instances with graphs of $N_V = 50, 100, 150, 200$ nodes.

It is easy to see that EA-FFP2 was capable of finding solutions close to quasi-optima also for benchmark FFP2 instances with $N_V = 100, 150, 200$. For many benchmark FFP2 instances, even the exact quasi-optimum was reached.

VI. Summary

This paper focused on an extension of the original Firefighter Problem (FFP), namely the Bi-Firefighter Problem

TABLE III: Results of EA-FFP2 on 50 benchmark FFP2 instances with graphs of 50 nodes. EA-FFP2 was run 10 times for each problem instance

			EA-FFP2						
		quasi optimum	min	max	mean	std	MAE	MAPE	# opt. reached
1	$N_b = N_f = 1, N_F = 5$	28.0229	28.0229	28.0229	28.0229	0.0000	0.00000000	0.00000000	10
2		33.6905	33.6905	33.6905	33.6905	0.0000	0.00000000	0.00000000	10
3		34.8573	34.8573	34.8573	34.8573	0.0000	0.00000000	0.00000000	10
4		19.6007	19.6007	19.6007	19.6007	0.0000	0.00000000	0.00000000	10
5		18.0539	18.0539	18.0539	18.0539	0.0000	0.00000000	0.00000000	10
6		34.4867	34.4867	34.4867	34.4867	0.0000	0.00000000	0.00000000	10
7		61.5614	61.5614	61.5614	61.5614	0.0000	0.00000000	0.00000000	10
8		21.3310	21.3310	21.3310	21.3310	0.0000	0.00000000	0.00000000	10
9		24.4558	24.4558	24.4558	24.4558	0.0000	0.00000000	0.00000000	10
10		26.0689	26.0689	26.0689	26.0689	0.0000	0.00000000	0.00000000	10
11	$N_b = N_f = 2, N_F = 5$	159.3318	161.9951	155.6224	156.2597	1.9118	3.07213971	0.01928139	9
12		43.7437	43.7437	43.7437	43.7437	0.0000	0.00000000	0.00000000	10
13		17.7697	17.7697	17.7697	17.7697	0.0000	0.00000000	0.00000000	10
14		44.6479	44.6479	44.6479	44.6479	0.0000	0.00000000	0.00000000	10
15		75.3712	75.3712	75.3712	75.3712	0.0000	0.00000000	0.00000000	10
16		102.3735	102.3735	102.3735	102.3735	0.0000	0.00000000	0.00000000	10
17		46.5432	46.5432	46.5432	46.5432	0.0000	0.00000000	0.00000000	10
18		56.7372	56.7372	56.7372	56.7372	0.0000	0.00000000	0.00000000	10
19		214.4774	219.5905	214.4774	214.9887	1.5339	0.51130883	0.00238398	9
20		49.1081	49.1081	49.1081	49.1081	0.0000	0.00000000	0.00000000	10
21	$N_b = N_f = 3, N_F = 5$	167.6293	167.6293	167.6293	167.6293	0.0000	0.00000000	0.00000000	10
22		90.5758	90.5758	90.5758	90.5758	0.0000	0.00000000	0.00000000	10
23		108.6657	102.9491	102.9491	102.9491	0.0000	5.71660427	0.05260724	10
24		146.1557	150.0418	146.1557	147.8246	1.6987	1.66892407	0.01141881	5
25		192.1975	192.1975	192.1975	192.1975	0.0000	0.00000000	0.00000000	10
26		117.4366	117.4366	117.4366	117.4366	0.0000	0.00000000	0.00000000	10
27		175.9925	175.9925	175.9925	175.9925	0.0000	0.00000000	0.00000000	10
28		69.2471	69.2471	69.2471	69.2471	0.0000	0.00000000	0.00000000	10
29		146.7491	146.7491	146.7491	146.7491	0.0000	0.00000000	0.00000000	10
30		178.1093	178.1093	178.1093	178.1093	0.0000	0.00000000	0.00000000	10
31	$N_b = N_f = 4, N_F = 5$	343.8124	343.8124	343.8124	343.8124	0.0000	0.00000000	0.00000000	10
32		148.1513	148.1513	148.1513	148.1513	0.0000	0.00000000	0.00000000	10
33		401.5236	401.5236	401.5236	401.5236	0.0000	0.00000000	0.00000000	10
34		237.0045	234.2337	234.2337	234.2337	0.0000	2.77073191	0.01169063	10
35		152.2785	155.9372	152.2785	153.0386	1.4517	0.76008614	0.00499142	7
36		352.1346	356.3095	340.2322	343.0979	5.7845	9.03666547	0.02566254	8
37		191.3792	191.3792	191.3792	191.3792	0.0000	0.00000000	0.00000000	10
38		139.6295	139.6295	139.6295	139.6295	0.0000	0.00000000	0.00000000	10
39		156.3796	156.3796	156.3796	156.3796	0.0000	0.00000000	0.00000000	10
40		85.1614	85.1614	85.1614	85.1614	0.0000	0.00000000	0.00000000	10
41	$N_b = N_f = 5, N_F = 5$	329.9183	329.9183	329.9183	329.9183	0.0000	0.00000000	0.00000000	10
42		304.8333	298.4630	298.4630	298.4630	0.0000	6.37029068	0.02089762	10
43		267.5602	267.7968	267.5602	267.6312	0.1084	0.07097939	0.00026528	7
44		434.4769	434.4769	434.4769	434.4769	0.0000	0.00000000	0.00000000	10
45		405.7045	405.7045	405.7045	405.7045	0.0000	0.00000000	0.00000000	10
46		304.3599	304.3599	304.3599	304.3599	0.0000	0.00000000	0.00000000	10
47		212.8050	212.8050	212.8050	212.8050	0.0000	0.00000000	0.00000000	10
48		327.2690	327.2690	327.2690	327.2690	0.0000	0.00000000	0.00000000	10
49		244.0910	244.0910	244.0910	244.0910	0.0000	0.00000000	0.00000000	10
50		217.9288	217.9288	217.9288	217.9288	0.0000	0.00000000	0.00000000	10

TABLE IV: Summary of results of EA-FFP2 on the benchmark FFP2 instances with graphs of $50, 100, 150, 200$ nodes. EA-FFP2 was run 10 times for each problem instance

			MAPE				
N_V	# of instances	# of runs	min	max	mean	std	# opt. reached
50	50	500	0.00000000	0.05260724	0.00298398	0.00904112	485
100	100	1000	0.00000000	0.33110408	0.03015175	0.04821550	821
150	150	1500	0.00000000	0.64492871	0.06495797	0.08289126	1450
200	200	2000	0.00000000	0.47473251	0.08367292	0.08736658	1956

(FFP2), where the second element (water) was introduced. An evolutionary approach to FFP2, the EA-FFP2 algorithm, based on a multi-modal Estimation of Distribution Algorithm (EDA), was proposed. It aimed at constructing a multi-modal probability distribution that was capable of generating efficient candidate solutions to the optimization problem. EA-FFP2 was validated on a number of benchmark FFP2 instances that were also solved by the branch and bound algorithms or the heuristic local search algorithms run for a large number of iterations for a long time. Computational experiments confirmed that EA-FFP2 was capable of solving FFP2 and finding solutions close to the optima determined by the branch and bound algorithms or to the quasi-optima determined by exhaustive local search.

ACKNOWLEDGMENT

This work was supported by the Polish National Science Centre (NCN) under grant no. 2015/19/D/HS4/02574. Calculations have been carried out using resources provided by Wroclaw Centre for Networking and Supercomputing (http://wcss.pl), grant no. 405.

REFERENCES

[1] Anshelevich, E., Chakrabarty, D., Hate, A., Swamy, C., Approximability of the firefighter problem, [in] Algorithmica, vol. 62, 2010, pp. 520-536.

[2] Anshelevich, E., Chakrabarty, D., Hate, A., Swamy, C., Approximation algorithms for the firefighter problem: cuts over time and submodularity, [in] Algorithms and Computation, LNCS, vol. 5878, Springer, 2009, pp. 974-983.

[3] Bazgan, C., Chopin, M., Ries, B., The firefighter problem with more than one firefighter on trees, [in] Discrete Applied Mathematics, vol. 161, 2013, pp. 899-908.

[4] Blum, C., Blesa, M., Garcia-Martinez, C., Rodriguez, F., Lozano, M., The firefighter problem: application of hybrid ant colony optimization algorithms, [in] Evolutionary Computation in Combinatorial Optimisation, LNCS, vol. 8600, Springer, 2014, pp. 218-229.

[5] Cai, L., Cheng, Y., Verbin, E., Zhou, Y., Surviving rates of graphs with bounded treewidth for the firefighter problem, [in] SIAM Journal on Discrete Mathematics, vol. 24, 2010, pp. 1322-1335.

[6] Cai, L., Wang, W., The surviving rate of a graph for the firefighter problem, [in] SIAM Journal on Discrete Mathematics, vol. 23, 2010, pp. 1814-1826.

[7] Costa, V., Dantas, S., Dourado, M. C., Penso, L., Rautenbach, D., More fires and more fighters, [in] Discrete Applied Mathematics, vol. 161, 2013, pp. 2410-2419.

[8] Develin, M., Hartke, S. G., Fire containment in grids of dimension three and higher, [in] Discrete Applied Mathematics, vol. 155, 2007, pp. 2257-2268.

[9] Esperet, L., van den Heuvel, J., Maffray, F., Sipma, F., Fire containment in planar graphs, [in] Journal of Graph Theory, vol. 73, 2013, pp. 267-279.

[10] Feldheim, O. N., Hod, R., 3/2 firefighters are not enough, [in] Discrete Applied Mathematics, vol. 161, 2013, pp. 301-306.

[11] Finbow, S., King, A., MacGillivray, G., Rizzi, R., The firefighter problem for graphs of maximum degree three, [in] Discrete Mathematics, vol. 307, 2007, pp. 2094-2105.

[12] Finbow, S., Macgillivray, G., The firefighter problem: a survey of results, directions and questions, [in] Australasian Journal of Combinatorics, vol. 43, 2009, pp. 57-77.

[13] Garcia-Martinez, C., Blum, C., Rodriguez, F., Lozano, M., The firefighter problem: Empirical results on random graphs, [in] Computers & Operations Research, vol. 60, 2015, pp. 55-66.

[14] Hartnell, B., Firefighter! an application of domination, [in] 20th Conference on Numerical Mathematics and Computing, 1995.

[15] Hu, B., Windbichler, A., Raidl, G. R., A new solution representation for the firefighter problem, [in] Evolutionary Computation in Combinatorial Optimization, LNCS, vol. 9026, Springer, 2015, pp. 25-35.

[16] Iwaikawa, Y., Kamiyama, N., Matsui, T., Improved approximation algorithms for firefighter problem on trees, [in] Transactions on Information and Systems, 2011, pp. 196-199.

[17] Larraanaga, P., Lozano, J., Estimation of Distribution Algorithms: a new tool for Evolutionary Computation, Kluwer Academic Publishers, 2001.

[18] Lipinski, P., A Hybrid Evolutionary Algorithm to Quadratic Three-Dimensional Assignment Problem with Local Search for Many-Core Graphics Processors, [in] Intelligent Data Engineering and Automated Learning, LNCS, vol. 6283, Springer, 2010, pp.343350.

[19] Lipinski, P., A Stock Market Decision Support System with a Hybrid Evolutionary Algorithm for Many-Core Graphics Processors, [in] Euro-Par 2010 Parallel Processing Workshops, LNCS, vol. 6586, Springer, 2011, pp. 455-462.

[20] Lipinski, P., Parallel Evolutionary Algorithms for Stock Market Trading Rule Selection on Many-Core Graphics Processors, [in] Natural Computing in Computational Finance, Studies in Computational Intelligence, vol. 380, Springer, 2012, pp.79-92.

[21] Messinger, M. E., Firefighting on infinite grids, MSc. Thesis, Department of Mathematics and Statistics, Dalhousie University, Halifax, Canada, 2004.

[22] Michalak, K., Auto-adaptation of genetic operators for multi-objective optimization in the firefighter problem, [in] Intelligent Data Engineering and Automated Learning, LNCS, vol. 8669, Springer, 2014, pp. 484-491.

[23] Michalak, K., The Sim-EA algorithm with operator autoadaptation for the multiobjective firefighter problem, [in] Evolutionary Computation in Combinatorial Optimization, LNCS, vol. 9026, Springer, 2015, pp. 184-196.

[24] Michalak, K., Knowles, J. D., Simheuristics for the multiobjective nondeterministic firefighter problem in a time-constrained setting, [in] Applications of Evolutionary Computation, LNCS, vol. 9598, Springer, 2016, pp. 248-265.