Simulation-Based Crossover for the Firefighter Problem

Krzysztof Michalak
Department of Information Technologies, Institute of Business Informatics, Wroclaw University of Economics
Komandorska 118-120
Wroclaw, Poland 53-345
krzysztof.michalak@ue.wroc.pl

ABSTRACT

The Firefighter Problem (FFP) is a combinatorial optimization problem in which the goal is to find the best way of protecting nodes in a graph from spreading fire or other threat given limited resources. Because of high computational complexity the FFP is often solved using metaheuristic methods such as evolutionary algorithms (EAs). The problem that arises is that many crossover operators used in EAs were developed with problems such as the TSP or the flowshop problem in mind and are therefore not optimized towards the FFP.

In this paper a new crossover operator SimX is proposed that determines how to combine information from parent specimens using a simulation of fire spreading and some problem-specific heuristics. The proposed crossover operator was compared to a set of ten permutation-based crossover operators in experiments performed using a multipopulation algorithm with operator autoadaptation. In the experiments it was observed that the proposed simulation-based crossover operator produced good offspring more often than the other operators.

CCS CONCEPTS

•**Mathematics of computing → Evolutionary algorithms;** •**Applied computing → Multi-criterion optimization and decision-making;**

KEYWORDS

Graph-based optimization, Multi-objective evolutionary optimization, Firefighter problem, Operator autoadaptation

ACM Reference format:
Krzysztof Michalak. 2017. Simulation-Based Crossover for the Firefighter Problem. In *Proceedings of GECCO '17, Berlin, Germany, July 15-19, 2017,* 9 pages.
DOI: http://dx.doi.org/10.1145/3071178.3071335

1 INTRODUCTION

The Firefighter Problem was proposed in 1995 by Hartnell [15] as an optimization problem in which fire spreads in discrete time steps in a graph and the goal is to protect the largest number of graph nodes possible. In each time step it is allowed to assign firefighters to a predefined number of nodes which become defended and are immune to fire until the end of the simulation. The same formulation can be used to describe spreading of diseases or activity of viruses in computer networks [24].

Works on the FFP can roughly be divided into three main areas. One is the theoretical analysis of the problem, aimed, for example, at understanding under what conditions it is possible to save the graph [11]. Another area is the application of classical optimization methods, such as the integer linear programming (ILP), to the FFP [6]. The ILP approach was augmented in some papers with simple problem-specific heuristics indicating which nodes to protect first [12]. The third area, which emerged only recently, concerns the application of metaheuristic methods to the FFP. The paper [3] concerning the application of the Ant Colony Optimization (ACO) to the FFP was published in 2014 and, according to its authors, was the first attempt to use a metaheuristic approach to the FFP. Also in 2014 a paper [18] was published in which the multiobjective version of the FFP was proposed and an evolutionary algorithm (EA) was used to tackle this version of the problem. At the EvoCOP conference in 2015 two more papers on the FFP were presented. One [16] in which the variable neighborhood search (VNS) was applied to the single-objective FFP and another [19] in which a multipopulation algorithm Sim-EA with migration based on similarity between search directions assigned to subpopulations was applied to the multiobjective version of the FFP. In all the papers mentioned above the spreading of fire was deterministic. The paper [12] studied random graphs, but the non-determinism was only present when the graphs were constructed and the starting points for the fire were selected - the spreading of fire in the graphs was deterministic. In 2016 a paper was published [20] in which an EA was applied to the non-deterministic FPP in which fire spread between connected nodes with some probability $P_{sp} < 1$. The approach used in the above mentioned paper is known as simheuristics (combination of a metaheuristic and simulation used for evaluating solutions) [17].

In this paper the algorithm from the paper [19] is used to test a newly proposed crossover operator on the deterministic FFP. In Section 2 a definition of the single-objective and the multiobjective version of the FFP is given. Section 3 discusses the proposed crossover operator. Section 4 presents the multipopulation algorithm with operator autoadaptation used to test the new crossover operator. Section 5 discusses the experiment plan and presents the results. Section 6 concludes the paper.

2 PROBLEM DEFINITION

In the Firefighter Problem spreading of fire is modelled on an undirected graph $G = \langle V, E \rangle$ with N_v vertices. The vertices from the set V can be in one of the states from a set $L = \{'B', 'D', 'U'\}$

GECCO '17, July 15-19, 2017, Berlin, Germany

ACM ISBN 978-1-4503-4920-8/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3071178.3071335

which are interpreted as $'B'$ = burning, $'D'$ = defended and $'U'$ = untouched. The spreading of fire is modelled in discrete time steps $t = 0, 1, \ldots$. In this paper we will use S_t to denote the state of the graph at time t and $S_t[v]$ to denote the state of vertex v at time t. Note, that in the FFP the state is only assigned to vertices. The edges define the topology and neither burn nor become defended by firefighters (if fire cannot spread along an edge it is because the vertex at the other end is defended). An instance of the FFP is an ordered triple $\langle G, S_0, N_f \rangle$ where G is the graph on which fire is modelled (represented, for example, as an adjacency matrix $A_{N_v \times N_v}$), S_0 is the initial state of the graph and N_f is the number of vertices that can become protected by firefighters in each time step. Most often, the initial state S_0 contains a few burning ($'B'$) vertices and the remaining ones are untouched ($'U'$). For $t > 0$ two events occur in each time step. First, firefighters are assigned to N_f still untouched ($'U'$) nodes, which become defended ($'D'$) and remain in this state until the end of the simulation. After firefighters are assigned, fire spreads along the edges of the graph from the burning nodes to untouched ones which change their state to $'B'$. Burning nodes ($'B'$) are considered lost and cannot be reclaimed by firefighters. The simulation ends when there are no untouched vertices adjacent to fire. A solution of the FFP describes an order in which firefighters are assigned to the nodes of graph G in consecutive time steps $t = 1, 2, \ldots$. Very often the permutation-based representation is used in which each solution is a permutation π of numbers $1, \ldots, N_v$, however, other methods of representing solutions have also been studied [16]. If a permutation π is a solution, the first N_f numbers for which the corresponding vertices are untouched ($'U'$) are taken from π in each time step and the firefighters are assigned to the nodes with these numbers.

The goal of the optimization is defined in a different way for the single-objective and multiobjective version of the FFP. In the **single-objective version** (used in the original FFP paper [15]) the goal is to maximize the number of non-burning nodes ('D' or 'U') at the simulation end. In the **multiobjective version** (proposed in [18]) there are m values $v_j(v)$, $j = 1, \ldots, m$ assigned to each node v in the graph. To calculate the objectives f_j, $j = 1, \ldots, m$ attained by a given solution π the spreading of fire is simulated starting from the initial state S_0 and ending at a certain time step t_{end}. After the simulation is finished, the objectives are calculated as:

$$f_j = \sum_{v \in V : S_{t_{end}}[v] \neq 'B'} v_j(v) \ , \qquad (1)$$

where $v_j(v)$ is the value of the node v according to the j-th criterion. In this paper the multiobjective version of the FFP is solved using the permutation-based solution representation.

3 SIMULATION-BASED CROSSOVER

In this paper the permutation-based solution representation is used in an EA solving the FFP. Typically, an evolutionary algorithm uses a crossover operator to spread information concerning good solutions in the population and a mutation operator to introduce new genetic information. In the literature numerous crossover operators are described that work on populations of chromosomes represented as permutations. However, many of those operators were developed with problems such as the TSP or the flowshop problem in mind and are therefore not optimized towards the FFP. Below, a new crossover operator SimX is described which uses simulation of the fire spreading to combine two permutation-based solutions to the FFP R_1 and R_2 into a new offspring O.

The simulation-based crossover operator uses two measures that can be calculated for each vertex v at a given graph state S_t. The first is $W(S_t, v)$ - the number of time steps it would take for the fire to reach a given untouched vertex v starting from the state S_t if no firefighters were assigned in these time steps. This value is equal to the length of the shortest path between any burning ($'B'$) vertex in state S_t and the vertex v. In this paper the value of $W(S_t, v)$ is calculated by simulating the spreading of fire starting from S_t and recording how many steps it takes the fire to reach each vertex v.

The second measure is a vertex degree $D(S_t, v)$ measured using only the edges of the graph G leading to vertices that are untouched in the state S_t. These measures are used to determine which vertex to select when a new offspring is constructed. Namely, those vertices are preferred, which are closer to fire (lower value of $W(S_t, v)$) and, in case the distances are the same, those that have the higher degree measured using $D(S_t, v)$. The usage of these particular heuristics to prioritize nodes for firefighter assignment is based on observations made in the literature [12, 20] where several heuristics for selecting nodes to defend were studied. The best heuristic from the ones tested in [20] turned out to be to place firefighters at the nodes adjacent to fire with highest degree.

The SimX crossover operator works by simulating the spreading of fire starting from the initial state S_0. In a typical FFP simulation at each time step $t = 1, 2, \ldots$ N_f firefighters are assigned to graph nodes which are determined using a permutation π. However, in the SimX crossover, the N_f firefighters to be assigned are not taken from a previously found solution, but are drawn from two parent solutions R_1 and R_2. These N_f firefighters are then appended to a list which records the choices made during the simulation and becomes a new offspring when the simulation ends. The working of the SimX crossover is presented in Algorithm 1. It consists mainly of a loop that simulates the spreading of fire in which several conditions are checked in order to determine which node to select. Note, that in Algorithm 1 when one of the conditions is used the remaining ones are skipped using the "continue" instruction.

In Algorithm 1 the following procedures are used.

- **Append(R_i, O, S)** - Removes the first node v from the list R_i, appends it at the end of the list O and marks the node v as defended in the state S (so that $S[v] = 'D'$). This procedure selects the first node in R_i as the one to assign firefighters to. This fact is reflected by adding this node to O and by protecting it in the graph.

- **Clip(R, S)** - Removes all nodes from the beginning of the given solution R for which the node state in S is different than $'U'$. This procedure ensures that the first element in R is an untouched node (in the graph state S).

- **Rand(P)** - Draws a random value from the given probability distribution P. In the SimX procedure it is used to obtain a random number drawn uniformly from the range $[0, 1]$.

Algorithm 1 The working of the SimX crossover.

IN: R_1, R_2 - parent solutions
OUT: O - a new offspring

$R'_1 := R_1$; $R'_2 := R_2$; $S := S_0$;
repeat
  for $f := 1, \ldots, N_f$ **do**
    Clip(R_1, S);
    Clip(R_2, S);
    // Stop if there are no more nodes to choose from
    if $R_1 = \emptyset$ **and** $R_2 = \emptyset$ **then**
      break;
    // If one of the parents is empty use the other one
    if $R_1 = \emptyset$ **then**
      Append(R_2, O, S);
      continue;
    if $R_2 = \emptyset$ **then**
      Append(R_1, O, S);
      continue;
    // If both parents are non-empty prefer the node closer to fire
    if $W(S, R_1[0]) < W(S, R_2[0])$ **then**
      Append(R_1, O, S);
      continue;
    if $W(S, R_2[0]) < W(S, R_1[0])$ **then**
      Append(R_2, O, S);
      continue;
    // If both parents are equally distant from fire prefer the node with a higher untouched degree
    if $D(S, R_1[0]) > D(S, R_2[0])$ **then**
      Append(R_1, O, S);
      continue;
    if $D(S, R_2[0]) > D(S, R_1[0])$ **then**
      Append(R_2, O, S);
      continue;
    // If both measures are equal select the source parent at random
    if Rand($U[0, 1]$)) < 0.5 **then**
      Append(R_1, O, S);
    else
      Append(R_2, O, S);
  // Stop if there are no more nodes to choose from
  if $R_1 = \emptyset$ **and** $R_2 = \emptyset$ **then**
    break;
  // Advance fire by one time step
  S' := SpreadFire(S);
  // Stop if the fire does not spread
  stop := ($S' = S$)
  $S := S'$;
until stop
Fill(O, R'_1)

- **Fill(O, R'_1)** - Fills the offspring O using those nodes in the original parent solution R'_1 that are not yet included in O. This procedure ensures that O is a valid permutation by adding the unused nodes at the end of O. This procedure works as follows:

  for $i := 1, \ldots, N_v$ **do**
      if $R'_1[i] \notin O$ **then**
          $O := [O, R'_1[i]]$;

- **SpreadFire(S)** - Creates a new graph state from the given state S, by spreading fire from burning nodes along graph edges to undefended nodes. In this procedure one time step is simulated, that is, fire only spreads for a distance of one edge.

4 OPTIMIZATION ALGORITHM

The optimization algorithm presented in this paper is based on the paper [19] in which a multipopulation evolutionary algorithm Sim-EA was used to solve the multiobjective version of the FFP. The Sim-EA algorithm uses migration between subpopulations based on a similarity measure. In the case of multiobjective optimization problems the Sim-EA works as a decomposition-based algorithm in which each subpopulation solves a subproblem obtained from the original one by aggregating the objectives using a different weights vector (see Figure 1). The similarity between subproblems is in this case expressed as a similarity of the directions of the weights vectors. The migration can be performed using different strategies, but in the previous works it was discovered that the strategy that worked best for the FFP was the rank-based strategy. In this strategy source subpopulations are ranked by similarity to the destination population with the least similar subpopulations assigned the lowest ranks. Then the roulette wheel selection is used to decide from which of the subpopulations to migrate specimens. The probability of selecting a subpopulation in this roulette wheel selection is proportional to the above mentioned rank, so the least similar subpopulations have the lowest probabilities.

The proposed algorithm has proven to have some advantages pertaining to the research on the FFP. In the paper [19] it produced better results on the FFP than a well-known MOEA/D algorithm using comparable computational resources. Also, the decomposition-based nature of the Sim-EA algorithm allows performing dynamic optimization in the case of the non-deterministic FFP [20] in which one has to select one, the best, solution from a population of non-dominated ones to act upon (i.e. assign firefighters before the next time step). The Sim-EA algorithm used in both [19] and [20] was equipped with an autoadaptation mechanism for selecting the crossover from a pool of 10 operators and mutation from a pool of 5 operators. The crossover operators used in these papers were: Cycle Crossover (CX) [22], Linear Order Crossover (LOX) [10], Merging Crossover (MOX) [21], Non-Wrapping Order Crossover (NWOX) [4], Order Based Crossover (OBX) [23], Order Crossover (OX) [13], Position Based Crossover (PBX) [23], Partially Mapped Crossover (PMX) [14], Precedence Preservative Crossover (PPX) [1, 2] and Uniform Partially Mapped Crossover (UPMX) [5]. The mutation operators were: displacement mutation, insertion mutation, inversion mutation, scramble mutation and transpose mutation. In this paper the same autoadaptation mechanism was used, except the SimX crossover was added to the pool of the crossover operators. The autoadaptation mechanism keeps track of the success rates of the operators (that is, how many times the operator produced a better solution than the parent solutions divided by the number of times this operator was used). These success rates reflect the effectiveness of the operators in generating good solutions. In the decomposition-based approach used in this paper the solutions are compared using the objectives aggregated using a weight vector assigned to the subpopulation. In the case of crossover operators each offspring O_1 and O_2 is compared to each parent P_1 and P_2, so the success rate of a crossover operator can potentially be as high as 4 (when O_1 improves on both P_1 and P_2 and O_2 likewise).

5 EXPERIMENTS AND RESULTS

The experiments were performed on graphs constructed according to the Erdős-Renyi model $G(N_v, P_{edge})$ introduced in [7–9]. In this model the number of vertices in the graph is N_v and for each pair of vertices an edge is added with a probability P_{edge} independently of other pairs. The Erdős-Renyi model was chosen, because it is one of the models most commonly studied in the literature. Experiments with other graph models were left for the further work.

In the experiments graphs with $N_v \in [50, 1250]$ were used, which is the range of graph sizes typically used in the literature [12, 16]. For $N_v \leq 250$ graphs prepared for the paper [19] were used. Each graph was generated by randomly determining with a probability $P_{edge} = 2.5/N_v$, for each pair of vertices v_i, v_j, if there exists an edge $\langle v_i, v_j \rangle$. This probability of generating an edge was used in order to ensure that the average number of edges adjacent to a vertex was similar for all the instances. For $N_v > 250$ the graphs were prepared in the same manner, but a probability $P_{edge} = 3.0/N_v$ was used, because for lower values these larger graphs tended to be easy to defend.

In the multiobjective FFP the vertices have m different costs which are used to calculate the m objectives. In this paper $m = 2$, so the solved problem is a biobjective one. The costs were generated by drawing pairs of random values with uniform probability on a triangle formed by points $[0, 0]$, $[0, 100]$, $[100, 0]$ which ensures that it is not possible to maximize both objectives at the same time, because the sum of costs associated with a vertex cannot exceed 100. Such setting enforces the algorithm to find good trade-offs between the objectives.

The parameters of the algorithm were set to exactly the same values as in the paper [19] in order to obtain comparable results. The number of subproblems was set to $N_{sub} = 20$ with the size of each of the subpopulations equal to the instance size N_v. Therefore, the total number of specimens used by the algorithm was $N_{spec} = 20 \cdot N_v$. The number of generations was set to $N_{gen} = 250$ generations for all problem instances. In this paper only the "rank" migration strategy that produced the best results in the paper [19] was used. For each graph size $N_{rep} = 30$ repetitions of the test were performed.

Success rates of the operators

In the experiments the success rate of each of the $N_{oper} = 11$ operators was recorded at each generation. In Figures 2-4 the values obtained for $N_v = 50$, 250 and 1000 are presented. These are

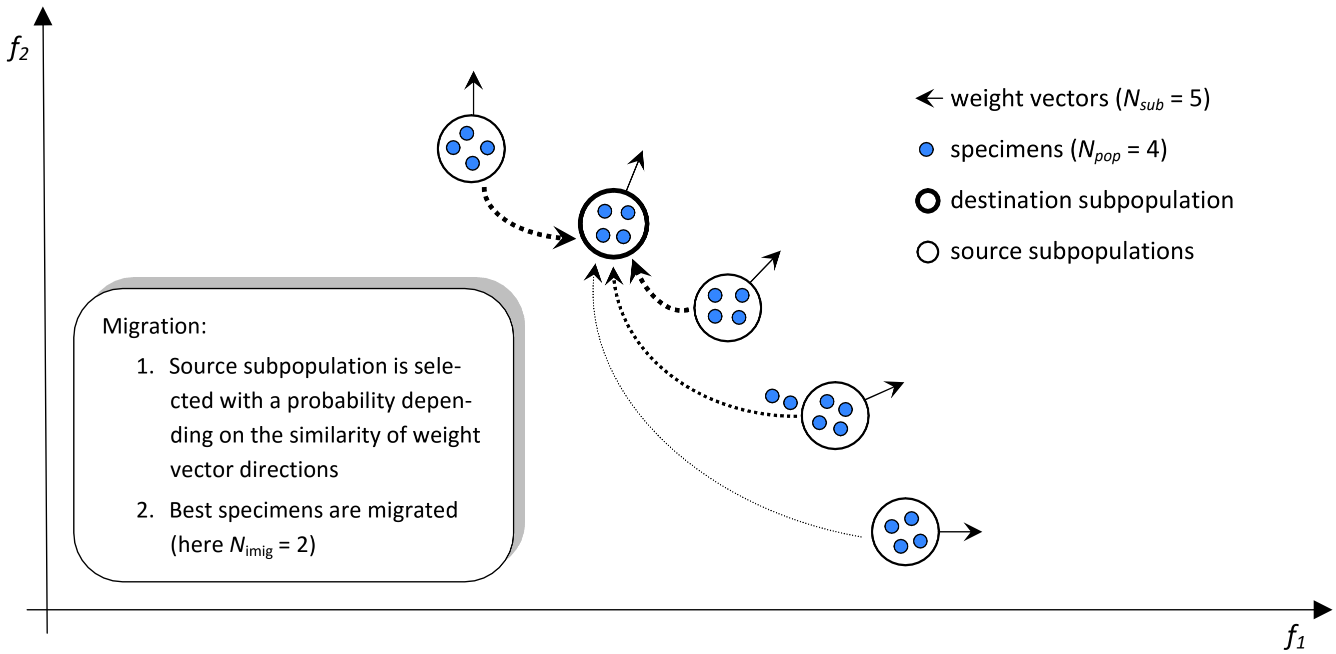

Figure 1: An overview of the key elements of the Sim-EA algorithm (see also [19]).

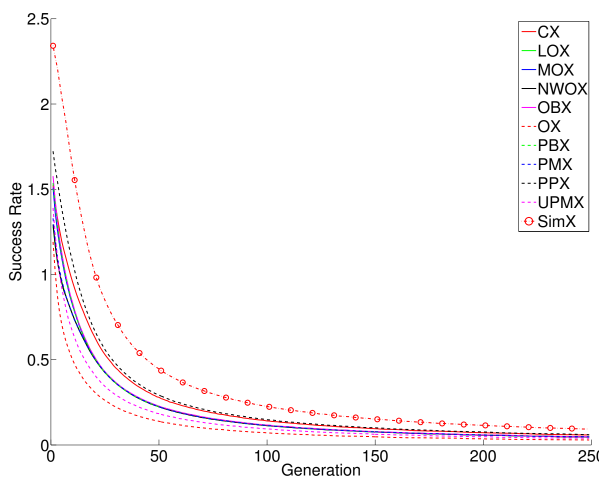

Figure 2: Success rates for the crossover operators for $N_v = 50$.

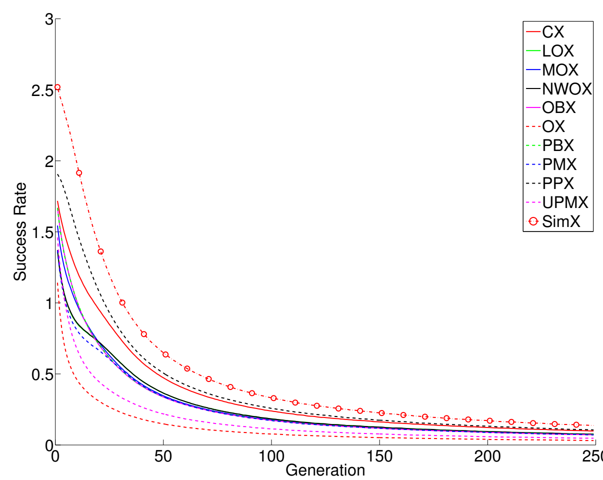

Figure 3: Success rates for the crossover operators for $N_v = 250$.

averages from the $N_{rep} = 30$ repetitions of the test plotted against the generation number for each of the operators ($N_{oper} = 11$ lines).

Due to limited space the plots for other values of N_v are omitted, however, the observed behaviour was very similar in all cases. To present the comparison between success rates of SimX and the other operators in a concise way the following procedure was adopted. The set of success rate values observed in the experiments forms a three dimensional array $X_{N_{gen} \times N_{oper} \times N_{rep}}$ (that is $X_{250 \times 11 \times 30}$). For each generation $g = 1, \ldots, N_{gen}$ we take a vector x formed from values observed at generation g in N_{rep} independent repetitions of the test for the SimX operator: $x = X[g, N_{oper}, :]$ (assuming that the results for the SimX are stored as the last ones - at the index N_{oper} and denoting by ":" the fact that we take the values for all test repetitions). Another vector x'_k is formed from values observed at generation g in N_{rep} independent repetitions of the test for each operator except SimX: $x'_k = X[g, k, :]$, where $k = 1, \ldots, N_{oper} - 1$.

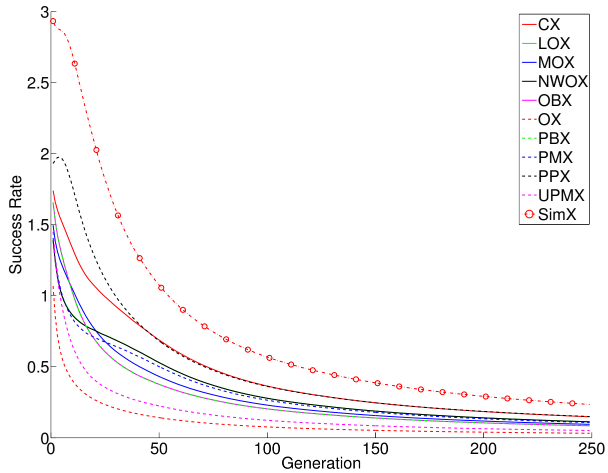

Figure 4: Success rates for the crossover operators for $N_v = 1000$.

For each k a comparison is performed between the k-th operator and the SimX. We calculate the difference between medians $\delta_{g,k} = \text{median}(x) - \text{median}(x'_k)$ and apply the Wilcoxon test to x and x'_k which has the null hypothesis of the equality of medians. Positive value of $\delta_{g,k}$ means that the SimX operator obtained better success rates than the k-th operator and if the p-value $p_{g,k}$ obtained from the Wilcoxon test is low we assume that the result is statistically significant (usually when the p-value is less than 0.05). To compare the SimX operator to all the 10 operators we calculate:

$$\delta_g^* = \min_{k=1,\ldots,N_{oper}-1} \delta_{g,k} \quad (2)$$

and

$$p_g^* = 1 - \prod_{k=1}^{N_{oper}-1} (1 - p_{g,k}) \ . \quad (3)$$

The equation (3) is a p-value corrected for family-wise error rate (FWER). In other words it is the probability that among 10 comparisons of the SimX to another operator we at least once mistakenly assumed that the medians are not equal. The value δ_g^* is the minimal difference between the median obtained using SimX and any other operator. Therefore, if δ_g^* is positive and p_g^* is low (e.g. < 0.05) we can state that at generation g the SimX operator outperformed all the others and this advantage is statistically significant. In order to summarize the results obtained in all generations the minimal and maximal values of δ_g^* and p_g^* obtained for each graph size N_v are presented in Table 1.

Clearly, in each round of tests in each generation the median success rate obtained for the SimX is larger than median success rates obtained for all other operators (minimal δ_g^* is always larger than 0). Also, the observed differences are statistically significant (the maximal value of p_g^* is never larger than 0.000075).

Final optimization results

Another question is if the higher success rates obtained by the SimX allow improving the optimization results. Because the algorithm used in this paper was the same as in [19] and the parameters

Table 1: Summary of the differences of medians δ_g^* and FWER-corrected p-values p_g^*.

N_v	δ_g^*		p_g^*	
	min	max	min	max
50	0.025000	0.655000	0.000016	0.000022
75	0.025000	0.650000	0.000016	0.000026
100	0.030000	0.575000	0.000017	0.000075
125	0.030000	0.575000	0.000014	0.000017
150	0.030000	0.600000	0.000016	0.000022
175	0.025000	0.615000	0.000016	0.000017
200	0.030000	0.515000	0.000016	0.000018
225	0.030000	0.550000	0.000024	0.000026
250	0.020000	0.685000	0.000016	0.000019
500	0.089260	1.105855	0.000017	0.000017
750	0.078677	1.103708	0.000017	0.000017
1000	0.077825	1.092556	0.000017	0.000017
1250	0.079126	0.992836	0.000017	0.000017

Table 2: Results obtained using 10 crossover operators and the same 10 crossover operators plus the SimX crossover. Results for 10 operators for $N_v \leq 250$ come from the paper [19].

N_v	10 operators	10 operators + SimX
50	621.2476	619.3017
75	714.5162	731.4931
100	742.6952	744.2601
125	792.7041	793.4963
150	857.3251	860.9700
175	838.2117	850.8454
200	854.1746	911.3520
225	948.5553	965.0240
250	962.4042	969.3159
500	2599.3692	2847.8968
750	2860.9900	2982.3727
1000	2979.1461	3162.9323
1250	3096.7110	3229.9015

and test data were the also the same, it was possible to compare the results obtained using the 10 crossover operators from the previous paper and the new set of operators including the SimX operator. In each of the 30 repetitions of the test after completing N_{gen} generations the best aggregated objective attained in each of the N_{sub} subproblems was recorded. Median values from these aggregated objectives are presented in Table 2.

Results presented in Table 2 show that, indeed, when the SimX crossover was used the results were improved in the case of all graphs with $N_v \geq 75$. Only for the smallest graphs ($N_v = 50$) the results were slightly deteriorated.

Dynamic behaviour

Apart from the final optimization results attained after completing all N_{gen} generations, it is possible to compare the optimization results with respect to the number of generations and running time.

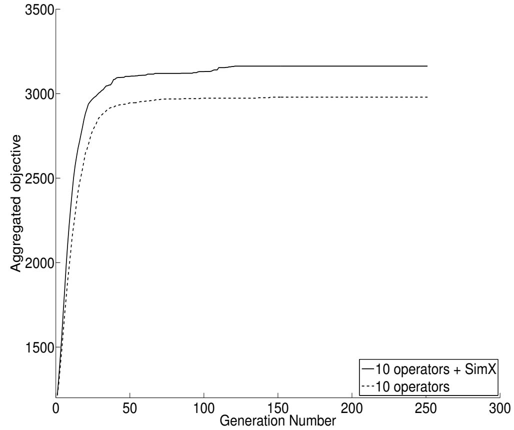

Figure 5: Median value of the aggregated objective for $N_v = 1000$ with respect to the number of generations.

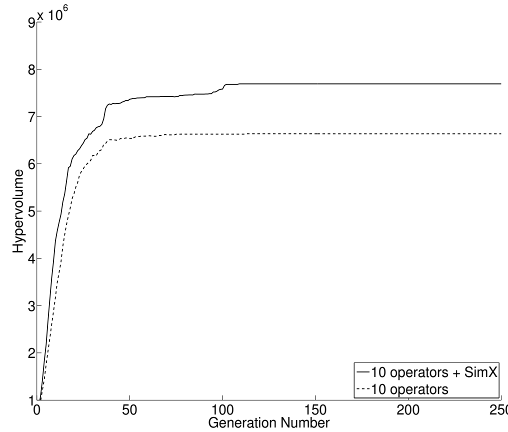

Figure 6: Median value of the hypervolume for $N_v = 1000$ with respect to the number of generations.

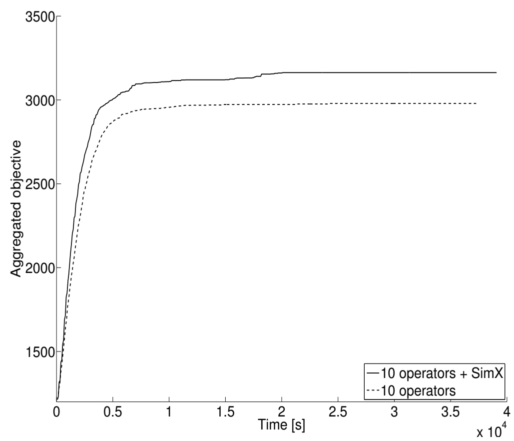

Figure 7: Median value of the aggregated objective for $N_v = 1000$ with respect to the running time.

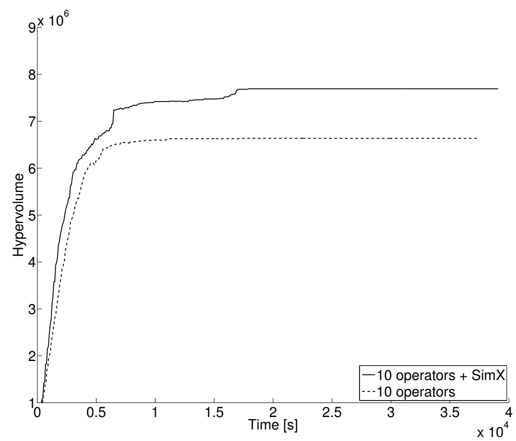

Figure 8: Median value of the hypervolume for $N_v = 1000$ with respect to the running time.

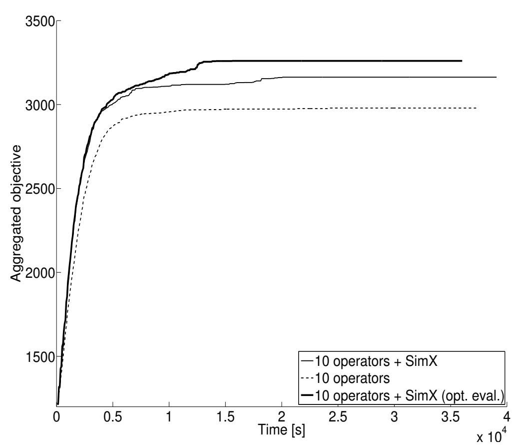

Figure 9: Median value of the aggregated objective for $N_v = 1000$ with respect to the running time (including the improved evaluation scheme).

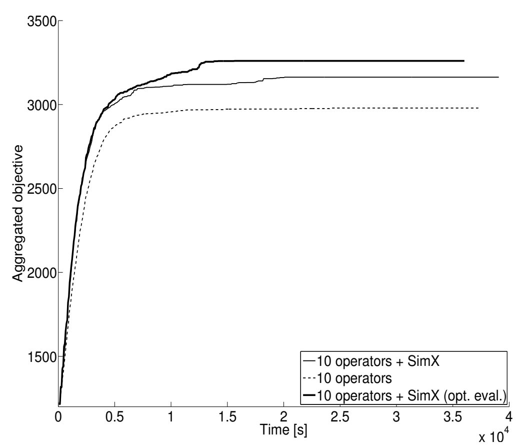

Figure 10: Median value of the hypervolume for $N_v = 1000$ with respect to the running time (including the improved evaluation scheme).

The latter comparison is important because, contrary to many other crossover operators, the SimX is computationally expensive.

In Figures 5 and 7 median values of the aggregated objective attained for $N_v = 1000$ are presented plotted with respect to the number of generations and running time. Figures 6 and 8 show median values of the hypervolume attained for $N_v = 1000$. From the presented figures it is clear that when the SimX operator is used the optimization results are improved compared to runs in which this operator is not used. In the case when the SimX is included, the graphs plotted with respect to the running time are shifted a little more to the right than those plotted with respect to the generation number reflecting the additional computational cost of the SimX. However, this effect is not large enough to negate the improvement of the results achieved by using the SimX.

Further improvement of the SimX operator

Because the simulation loop used for constructing new solutions in Algorithm 1 is very similar to the simulation loop used for evaluating solutions it is possible to calculate the objectives using the equation (1) right after Algorithm 1 finishes. Incorporating this approach in an evolutionary algorithm requires only a slight modification, taking into account the fact that specimen evaluation is performed by the SimX internally, but has to be done separately for the other crossover operators. The results obtained using the above-mentioned approach are presented in Figures 9 and 10 with graphs plotted with respect to the running time of the algorithm. Clearly, better results are obtained in the same amount of time when the simulation loop in SimX is used for calculating the objectives, compared to the situation when solution evaluation is performed separately by the evolutionary algorithm.

6 CONCLUSION

In this paper a new crossover operator SimX was proposed which is dedicated to the Firefighter Problem. This operator uses simulation of the spreading of fire to decide in what order the elements of parent solutions are copied to the offspring. The experiments shown that the proposed operator attained higher values of success rate (that measures how often the offspring represent better solutions than parents) than other 10 crossover operators used for comparison. Also, when the SimX operator was used in addition to the other operators the optimization results were improved.

An interesting direction for the further work on the SimX operator would be to test the working of this operator on other graph topologies. The Erdős-Renyi model used in this paper is very common in the literature, but there are others, such as scale-free graphs, which are representative of systems occurring in real life.

ACKNOWLEDGMENT

This work was supported by the Polish National Science Centre 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. 407.

REFERENCES

[1] Christian Bierwirth, Dirk C. Mattfeld, and Herbert Kopfer. 1996. On Permutation Representations for Scheduling Problems. In *Proceedings of the 4th Internat. Conf. on Parallel Problem Solving from Nature*. Springer-Verlag, 310–318.

[2] Joe L. Blanton, Jr. and Roger L. Wainwright. 1993. Multiple Vehicle Routing with Time and Capacity Constraints Using Genetic Algorithms. In *Proceedings of the 5th Internat. Conf. on Genetic Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 452–459.

[3] Christian Blum, M. J. Blesa, C. Garcia-Martinez, F. J. Rodriguez, and M. Lozano. 2014. The Firefighter Problem: Application of Hybrid Ant Colony Optimization Algorithms. In *Evolutionary Computation in Combinatorial Optimisation*, Christian Blum and Gabriela Ochoa (Eds.). LNCS, Vol. 8600. Springer Berlin Heidelberg, 218–229.

[4] Vincent A. Cicirello. 2006. Non-wrapping Order Crossover: An Order Preserving Crossover Operator That Respects Absolute Position. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*. ACM, New York, NY, USA, 1125–1132.

[5] V. A. Cicirello and S. F. Smith. 2000. Modeling GA performance for control parameter optimization. In *GECCO-2000: Proceedings of the Genetic and Evolutionary Computation Conference: A Joint Meeting of the Ninth Internat. Conf. on Genetic Algorithms (ICGA-2000) and the Fifth Annual Genetic Programming Conference (GP-2000), July 10-12, 2000, Las Vegas, Nevada*, L.D. Whitley (Ed.). Morgan Kaufmann Publishers, 235–242.

[6] Mike Develin and Stephen G. Hartke. 2007. Fire Containment in Grids of Dimension Three and Higher. *Discrete Appl. Math.* 155, 17 (2007), 2257–2268.

[7] P. Erdős. 1947. Some remarks on the theory of graphs. *Bull. Amer. Math. Soc.* 53, 4 (1947), 292–294.

[8] P. Erdős and A. Rényi. 1959. On random graphs. I. *Publ. Math. Debrecen* 6 (1959), 290–297.

[9] P. Erdős and A. Rényi. 1960. On the Evolution of Random Graphs. In *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, Vol. 5. 17–61.

[10] E. Falkenauer and S. Bouffouix. 1991. A Genetic Algorithm for Job Shop. In *Proceedings of the 1991 IEEE Internat. Conf. on Robotics and Automation*. 824–829.

[11] Ohad N. Feldheim and Rani Hod. 2013. 3/2 firefighters are not enough. *Discrete Applied Mathematics* 161, 1-2 (2013), 301–306.

[12] C. Garca-Martnez, C. Blum, F.J. Rodriguez, and M. Lozano. 2015. The firefighter problem: Empirical results on random graphs. *Computers & Operations Research* 60 (2015), 55–66.

[13] D. Goldberg. 1989. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley.

[14] David E. Goldberg and Robert Lingle Jr. 1985. Alleles, loci, and the traveling salesman problem. In *Proceedings of the First Internat. Conf. on Genetic Algorithms and Their Applications*, John J. Grefenstette (Ed.). Lawrence Erlbaum Associates, Publishers, 154–159.

[15] B. Hartnell. 1995. Firefighter! An application of domination. In *20th Conference on Numerical Mathematics and Computing*.

[16] Bin Hu, Andreas Windbichler, and Gnther R. Raidl. 2015. A New Solution Representation for the Firefighter Problem. In *Evolutionary Computation in Combinatorial Optimization*, Gabriela Ochoa and Francisco Chicano (Eds.). LNCS, Vol. 9026. Springer, 25–35.

[17] Angel A. Juan, Javier Faulin, Scott E. Grasman, Markus Rabe, and Gonalo Figueira. 2015. A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives* 2 (2015), 62–72.

[18] Krzysztof Michalak. 2014. Auto-adaptation of Genetic Operators for Multi-objective Optimization in the Firefighter Problem. In *Intelligent Data Engineering and Automated Learning - IDEAL 2014*, Emilio Corchado, Jos A. Lozano, Hctor Quintin, and Hujun Yin (Eds.). LNCS, Vol. 8669. Springer, 484–491.

[19] Krzysztof Michalak. 2015. The Sim-EA Algorithm with Operator Autoadaptation for the Multiobjective Firefighter Problem. In *Evolutionary Computation in Combinatorial Optimization*, Gabriela Ochoa and Francisco Chicano (Eds.). LNCS, Vol. 9026. Springer, 184–196.

[20] Krzysztof Michalak and Joshua D. Knowles. 2016. Simheuristics for the Multiobjective Nondeterministic Firefighter Problem in a Time-Constrained Setting. In *Applications of Evolutionary Computation: 19th European Conference, EvoApplications 2016, Porto, Portugal, March 30 – April 1, 2016, Proceedings, Part II*, Giovanni Squillero and Paolo Burelli (Eds.). 248–265.

[21] Christine L. Mumford. 2006. New Order-based Crossovers for the Graph Coloring Problem. In *Proceedings of the 9th Internat. Conf. on Parallel Problem Solving from Nature*. Springer-Verlag, Berlin, Heidelberg, 880–889.

[22] I. M. Oliver, D. J. Smith, and J. R. C. Holland. 1987. A Study of Permutation Crossover Operators on the Traveling Salesman Problem. In *Proceedings of the Second Internat. Conf. on Genetic Algorithms on Genetic Algorithms and Their Applications*. Lawrence Erlbaum Associates Inc., Hillsdale, NJ, USA, 224–230.

[23] Gilbert Syswerda. 1991. Schedule Optimization Using Genetic Algorithms. In *Handbook of Genetic Algorithms*, Lawrence Davis (Ed.). Van Nostrand Reinhold, New York, NY.

[24] Weifan Wang, Stephen Finbow, and Ping Wang. 2010. The surviving rate of an infected network. *Theoretical Computer Science* 411, 40..42 (2010), 3651–3660.