The Sim-EA Algorithm with Operator Autoadaptation for the Multiobjective Firefighter Problem

Removed due to the anonymization requirement

Removed due to the anonymization requirement
Removed

Abstract. The firefighter problem is a graph-based optimization problem that can be used for modelling the spread of fires, and also for studying the dynamics of epidemics. Recently, this problem gained interest from the softcomputing research community and papers were published on applications of ant colony optimization and evolutionary algorithms to this problem. Also, the multiobjective version of the problem was formulated.

This paper proposes a multipopulation algorithm Sim-EA for the multiobjective version of the firefighter problem. The algorithm optimizes firefighter assignment for a predefined set of weight vectors which determine the importance of individual objectives. A migration mechanism based on similarity of search directions is used for improving the effectiveness of the algorithm.

Obtained results confirm that the multipopulation approach works better than the decomposition approach in which a single specimen is assigned to each direction. Given less computational resources than the decomposition approach, the Sim-EA algorithm produces better results. The migration strategy based on ranking of subpopulations according to similarity of search directions seems to work best among the migration strategies tested in this paper.

Keywords: multipopulation algorithms, multi-objective evolutionary optimization, graph-based optimization, firefighter problem

1 Introduction

The firefighter problem is a graph-based optimization problem formalized in 1995 by Hartnell [16]. This problem can be used for modelling the spread of fires, and also for studying the dynamics of epidemics. The original version of the firefighter problem was single objective. Recently [1], a multiobjective version of the firefighter problem has been proposed.

Developments concerning the firefighter problem can be divided into several areas. Some papers deal with theoretical properties and discuss specific types of graphs and specific problem cases. For example the paper [13] gives lower and upper bounds on the amount of firefighters needed to control a fire in the case of specific planar grids.

Some attempts on this problem were also made by the optimization research community. For example, the paper [11] uses a linear integer programming model to solve the single-objective version of the problem. The applications of metaheuristic methods to the firefighter problem are relatively recent. The paper [6] published in 2014 applies an Ant Colony Optimization (ACO) approach to the single-objective version of the firefighter problem. The authors of this paper have stated, that before its publication not a single metaheuristic approach has been applied to this problem. In the paper [1], published later in 2014 a multiobjective version of the firefighter is tackled using the NSGA-II algorithm [9] with an autoadaptation mechanism selecting the best performing genetic operators.

In this paper the multiobjective version of the problem is approached, however, it is treated a little differently than in [1]. Instead of finding the Pareto front we are interested in finding optimal firefighter assignments for a predefined set of N_{sub} weight vectors $W_0 = \left\{\lambda^{(1)}, \ldots, \lambda^{(N_{sub})}\right\}$ which determine the directions of the search as follows. Denote the number of objectives as m. A vector $\lambda^{(j)} = \left[\lambda_1^{(j)}, \ldots, \lambda_m^{(j)}\right]$ determines a search direction in which the first objective has weight $\lambda_1^{(j)}$, the second $\lambda_2^{(j)}$ and so on. For example, a search along direction determined by the vector $\lambda = [1/3, 2/3]$ generates strategies which are best suited if we are interested twice as much in maximizing objective f_2 as in maximizing f_1. This is the approach that is employed in decomposition-based algorithms such as the MOEA/D [17, 23]. The approach presented in this paper is intended to be used in research concerning dynamic optimization in the case of the non-deterministic version of the firefighter problem which is currently underway. In the case of dynamic optimization the algorithm may modify the firefighter assignment in response to changes in the environment. This in turn affects the state of the environment. When a decomposition-based approach is used one can simulate the interaction between the algorithm and the environment for each optimization direction separately. Therefore, it is possible to assess in advance, what will happen if certain weights are assigned to the objectives and it is possible to compare various scenarios. To use a Pareto-based algorithm one would have to employ a decision-making module which would choose which strategy should interact with the simulation. Therefore, in this paper a decomposition-based approach is adopted and, instead of the NSGA-II algorithm as in [1], a multipopulation algorithm is used in which subpopulations optimize the solutions along predefined weight vectors from the set W_0. The algorithm used in this paper is based on the previous work [2] which introduced Sim-EA – a multipopulation algorithm with migration based on problem instance similarities. In this paper a different similarity measure than in [2] is used which can be calculated based on the weight vectors from the set W_0 and a new rank-based migration strategy is proposed. The multipopulation algorithm is combined with the operator autoadaptation mechanism used in [1].

To sum up, while this paper uses some of the techniques presented in previous works it introduces the following new elements. Compared to the previous work on the multiobjective firefighter problem [1] the approach was changed from a Pareto-based to a decomposition-based optimization. Following that a different algorithm was employed. With respect to [2] the obvious difference is that this paper tackles a different problem. From this follows that a different way of determining subproblem similarity had to be used. Also, an operator autoadaptation mechanism was used in this paper and a new migration strategy was proposed.

The rest of the paper is structured as follows. In Section 2 the single and multiobjective versions of the firefighter problem are defined. Section 3 presents the algorithm proposed for solving the multiobjective firefighter problem. Section 4 describes the experimental setup and presents the results. Section 5 concludes the paper.

2 Problem Definition

The single objective version of the firefighter problem can be formalized as follows. An undirected graph $G = \langle V, E \rangle$ with N_v vertices is given. Each vertex of the graph G can be labeled using labels from the set $L = \{'B',' D',' U'\}$ with the interpretation $'B'$ = burning, $'D'$ = defended and $'U'$ = untouched. Further, we will use a function $l : V \rightarrow L$ to denote the labelling of the vertices of the graph G. The spread of fire is simulated in discrete time steps. Initially, vertices from a non-empty set $\emptyset \neq S \subset V$ are labelled 'B' and the remaining ones are labelled 'U'. In each of the following steps of the simulation two events occur. First, a predefined number N_f of still untouched nodes of the graph G (labelled $'U'$) become defended by firefighters (i.e. become labelled $'D'$). Then the fire spreads from the nodes labelled $'B'$ to all the neighbouring nodes labelled $'U'$. The nodes marked $'D'$ remain defended until the end of the simulation and the fire cannot spread to them. The simulation is stopped either when the fire is contained (i.e. there are no nodes labelled $'U'$ to which the fire could spread via the edges of the graph G) or when all the undefended nodes are burning.

The order in which firefighters are assigned can be represented as a permutation P of the numbers $1, \ldots, N_v$. When firefighters are assigned the first N_f yet untouched nodes are taken from the permutation P. Thus, in every time step exactly N_f firefighters are assigned (with the exception of the final time step in which the number of the untouched nodes may be less than N_f).

In the single objective version of the problem the goal is to find such permutation P that maximizes the number of non-burning nodes at the end of the simulation. In the multiobjective version there are m values $v_i(v)$, $i = 1, \ldots, m$ assigned to each node v in the graph. Each of the v_i values for a given node v represents the value of this node with respect to a certain criterion. In the context of fire containment these criteria could represent, for example, the financial value $v_1(v)$ and the cultural importance $v_2(v)$ of the items stored at the node v.

To calculate the values of the objectives f_i, $i = 1, \ldots, m$ attained by a given permutation P one has to perform the simulation of the fire spread. After the simulation is finished, the values of the objectives are calculated as:

$$f_i = \sum_{v \in V : l(v) \neq' B'} v_i(v) \quad (1)$$

where:
$v_i(v)$ is the value of a given node according to the i-th criterion.

As mentioned in the introduction, the approach adopted in this paper is a decomposition-based optimization. Therefore, we actually search for a set of N_{sub} solutions each of which is the best possible one for a subproblem in which the objective function is an aggregation of the original objectives parameterized by a certain weight vector $\lambda^{(j)} = \left[\lambda_1^{(j)}, \ldots, \lambda_m^{(j)}\right]$, where $j = 1, \ldots, N_{sub}$.

3 The Sim-EA Algorithm

The Sim-EA algorithm presented in this paper is a multipopulation approach. Instead of forming one big population the specimens are divided into N_{sub} subpopulations, each consisting of N_{pop} specimens and a migration mechanism is used to facilitate information transfer between subpopulations as presented in Figure 1.

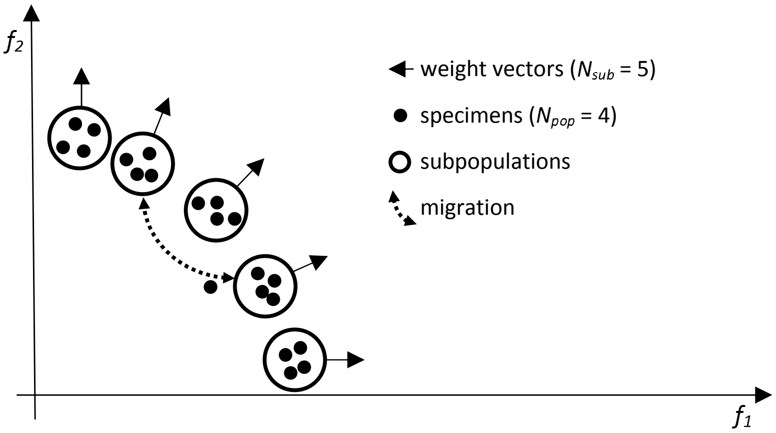

Fig. 1. An overview of elements of the Sim-EA algorithm.

The general outline of the algorithm is based on the previous work [2] in which the Sim-EA algorithm was used for optimizing several similar instances of the TSP problem at once. The overview of the Sim-EA algorithm is presented in Algorithm 1.

3.1 Migration

The key element of the Sim-EA algorithm is the migration mechanism which is based on the similarity of problem instances solved by the subpopulations.

Algorithm 1 The overview of the Sim-EA algorithm [2].

```
IN:
    $N_{gen}$ - the number of generations
    $N_{pop}$ - the size of each subpopulation
    $N_{sub}$ - the number of subpopulations
    $N_{imig}$ - the number of migrated specimens

Calculate the problem instance similarity matrix $S_{[N_{sub} \times N_{sub}]}$
Initialize subpopulations $P_1, P_2, \ldots, P_{N_{sub}}$.
for $g = 1, \ldots, N_{gen}$ do
    Apply genetic operators
    for $d = 1, \ldots, N_{sub}$ do
        $s = \underset{t}{\mathrm{argmax}}(S_{d,t})$
        $P'_d$ = the $N_{imig}$ best specimens from $P_s$
    end for
    for $d = 1, \ldots, N_{sub}$ do
        for $x \in P'_d$ do
            $P'_d = P'_d - \{x\}$
            w = the weakest specimen in $P_d$
            $P_d = P_d - \{w\}$
            b = BinaryTournament(w, x)
            $P_d = P_d \cup \{b\}$
        end for
    end for
    Apply genetic operators
    for $d = 1, \ldots, N_{sub}$ do
        e = the best specimen in $P_d$
        $P_d$ = Select($P_d \backslash \{e\}$, $N_{pop} - 1$)
        $P_d = P_d \cup \{e\}$
    end for
end for
```

In the previous work this similarity was measured using squared differences between cost matrices used in TSP instances. In this paper all subpopulations solve the same instance of the firefighter problem (i.e. they work on the same graph G), but have objective functions aggregated using different weight vectors $\lambda^{(j)}$. Consequently, the similarity $S_{i,j}$ between subproblems parameterized by weight vectors $\lambda^{(i)}$ and $\lambda^{(j)}$ is calculated as the dot product:

$$S_{i,j} = \lambda^{(i)} \cdot \lambda^{(j)} \quad . \tag{2}$$

The migration is performed in two phases. First, for each subpopulation P_d, $d = 1, \ldots, N_{sub}$ a source population P_s is determined according to the migration strategy. A set of immigrants P'_d is built by selecting the N_{imig} best specimens from the population P_s. In the second phase the immigrants from the sets P'_d are merged into the populations P_d for $d = 1, \ldots, N_{sub}$. During the merge phase each immigrant participates in a binary tournament [18] against the currently weakest specimen in the existing population P_d. If the immigrant has higher value of the aggregated objective function it replaces the weakest specimen in the population P_d.

The selection of the source subpopulations P_s depends on the adopted migration strategy. In this paper one new migration strategy named **"rank"** is proposed. In this strategy all subpopulations $P_{s'}$ except P_d are ranked according to the increasing values of the dot product $S_{d,s'} = \lambda^{(d)} \cdot \lambda^{(s')}$. From them one subpopulation P_s is selected using roulette wheel selection with probabilities proportional to the ranks of the subpopulations. Also, the three migration strategies introduced in the previous work were used in the tests:

"nearest" in which the source population P_s is the one with the highest value of the dot product (2), **"uniform"** in which the source population P_s is selected randomly with uniform probability distribution among all populations except P_d and **"none"** in which no migration is performed.

3.2 Operator autoadaptation

In the previous paper [1] in which the multiobjective firefighter problem was tackled using the NSGA-II algorithm an operator autoadaptation mechanism was used. From the results presented in the aforementioned paper it follows that various operators (e.g. different crossover procedures) perform well on various stages of the search. Therefore, in this paper an autoadaptation mechanism was also used. The autoadaptation mechanism used in this paper is based on success rates of the genetic operators. It counts the number of times each operator was used n_i and the number of improvements obtained b_i. Since in this paper a decomposition-based approach is used the improvements are determined with respect to the aggregated objective function not with respect to the individual objectives. However, after one application of the crossover operator the value of b_i can increase by more than 1, because each offspring improving on each parent is counted separately. Thus, a maximum increase of 4 can be obtained (both offspring improving on both parents). From the n_i and b_i values the success rate $s_i = b_i/n_i$ is calculated (if $n_i = 0$ then $s_i = 0$).

The s_i values are used to calculate probabilities with which individual operators are used. A minimum probability P_{min} is given to each of the N_{op} operators and the remaining $1 - N_{op}P_{min}$ is divided proportionally to success rates s_i of the operators. When an operator is to be applied a roulette wheel selection is performed. The selection of crossover and mutation operators is performed separately.

In this paper a set of 10 crossover operators and 5 mutation operators was used for genetic operations. The crossover operators were: Cycle Crossover (CX) [20], Linear Order Crossover (LOX) [12], Merging Crossover (MOX) [3, 19], Non-Wrapping Order Crossover (NWOX) [8], Order Based Crossover (OBX) [21], Order Crossover (OX) [14], Position Based Crossover (PBX) [21], Partially Mapped Crossover (PMX) [15], Precedence Preservative Crossover (PPX) [5, 4] and Uniform Partially Mapped Crossover (UPMX) [7]. The mutation operators were:

displacement mutation, insertion mutation, inversion mutation, scramble mutation and transpose mutation.

4 Experiments and Results

In the experiments the Sim-EA algorithm with various migration strategies was tested. Also, a single population approach known from the MOEA/D algorithm was used for comparison. The MOEA/D algorithm used the same operators and the autoadaptation mechanism as the Sim-EA.

Test data sets were prepared as follows for graph size $N_v = 50$, 75, 100, 125, 150, 175, 200, 225 and 250. The graph G was generated by randomly determining, for each pair of vertices v_i, v_j, if there exists an edge $\langle v_i, v_j \rangle$. The probability of generating an edge was set to $P_{edge} = 2.5/N_v$ in order to ensure that the average number of edges adjacent to a vertex was similar for all the instances. Costs were assigned to all vertices of the graph G by drawing pairs of random values with uniform probability on a triangle formed by points $[0, 0]$, $[0, 100]$, $[100, 0]$. Such an assignment 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.

In the experiments the performance of the Sim-EA algorithm with four migration strategies (nearest, rank, uniform and none) and the MOEA/D algorithm with three different parameter settings was tested. The number of subproblems was set to $N_{sub} = 20$.

The Sim-EA algorithm was parameterized by setting the size of each of the subpopulations to the instance size N_v. Therefore, the total number of specimens used by the Sim-EA algorithm was $N_{spec} = 20 \cdot N_v$. The number of generations was set to $N_{gen} = 250$ generations for all problem instances.

The MOEA/D algorithm does not use subpopulations, therefore the parameter setting had to be different. The size of the population is determined by a parameter H - the size of a step used for generating weight vectors. Because there is exactly one weight vector assigned to each specimen the population size is equal to the number of weight vectors. For an m-objective problem, weight vectors $\lambda^{(j)}$ are generated by selecting all the possible m-element combinations of numbers from the set:

$$\left\{ \frac{0}{H}, \frac{1}{H}, \ldots, \frac{H}{H} \right\} \tag{3}$$

such that:

$$\sum_{i=1}^{m} \lambda_i^{(j)} = 1. \tag{4}$$

The number of weight vectors and at the same time the number of specimens is:

$$N_{spec} = C_{H+m-1}^{m-1} \ . \tag{5}$$

For a biobjective optimization problem ($m = 2$) the set of weight vectors is $W = \{\lambda^{(j)}\}$, $j \in \{0, \dots, H\}$, where:

$$\lambda^{(j)} = \left[\frac{j}{H}, \frac{H-j}{H} \right]. \tag{6}$$

Because in this paper we are interested in finding optimal solutions along $N_{sub} = 20$ predefined directions the number of specimens in the experiments had to be chosen in such a way, that each of the N_{sub} weight vectors is present in the set W (i.e. $W_0 \subset W$). In cases when the population size N_{spec} was larger than N_{sub} additional N_i weight vectors were added between the main N_{sub} predefined weight vectors as presented in Figure 2.

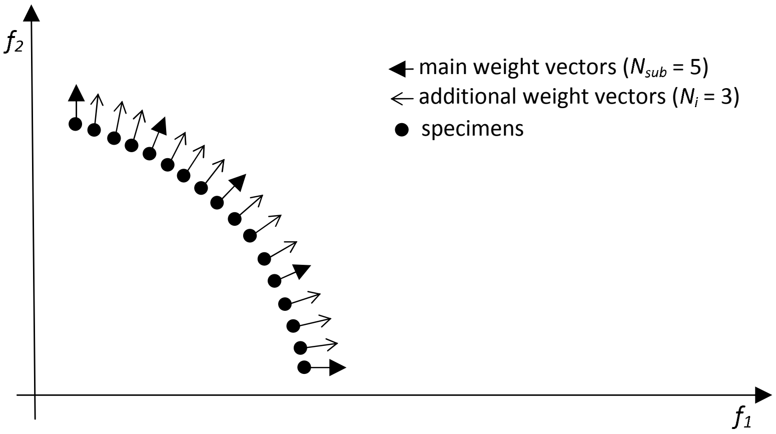
f2
main weight vectors (N_{sub} = 5)
additional weight vectors (N_i = 3)
● specimens
f1

Fig. 2. The configuration of weight vectors in the MOEA/D algorithm used for optimization along $N_{sub} = 5$ main directions with the total population size $N_{spec} = 17$.

Three different sets of parameters were used with the MOEA/D algorithm (cf. Table 1). In the first parametrization $MOEA/D_{20}$ the number of specimens was 20 for all data sets (i.e. $N_{spec} = N_{sub}$). In the two remaining parameter sets a larger population was used. From this larger population $N_{sub} = 20$ specimens were associated with weight vectors from the set W_0 which represented the main search directions (i.e. the aggregations for which the best solutions had to be found). The remaining specimens were associated with search directions between the main N_{sub} vectors. In the $MOEA/D_{210}$ parametrization $N_i = 10$ specimens were added between each two main search directions, thus the total population size was 210. In the $MOEA/D_{N_{spec}}$ parameter set the total number of specimens in the MOEA/D algorithm was set to a number a little larger than the total number of specimens N_{spec} used in the Sim-EA algorithm. To achieve this a varying number N_i of additional specimens were used. To make a fair comparison, the number of generations for each MOEA/D parameter set was adjusted in such a way that the minimum runtime of the MOEA/D in each repetition of the test was sligthly larger than the maximum runtime of the Sim-EA algorithm.

Table 1. Parameter sets used in the experiments with MOEA/D algorithm.

N_v	$MOEA/D_{20}$			$MOEA/D_{210}$			$MOEA/D_{N_{spec}}$		
	N_{spec}	N_{gen}	N_i	N_{spec}	N_{gen}	N_i	N_{spec}	N_{gen}	N_i
50	20	400	0	210	200	10	1008	50	52
75		1000			500		1502	70	78
100		2000			750		2015	80	105
125		3000			1000		2509	90	131
150		5000			1400		3003	100	157
175		7000			1800		3516	110	184
200		10000			2700		4010	150	210
225		12000			2800		4504	160	236
250		15000			3600		5017	200	263

For each problem instance 30 repetitions of the test were performed for each of the four migration strategies and for each parameter set used with the MOEA/D algorithm. After completing N_{gen} generations the best result obtained along each of the N_{sub} directions was recorded. From the gathered results median values obtained in the 30 iterations were calculated. The results for all test instances are summarized in Table 2. For each data set the best result is underlined. The results are also presented in Figure 3.

Table 2. Median values of the aggregated objective functions obtained in the experiments.

N_v	Sim-EA				MOEA/D		
	none	nearest	rank	uniform	$MOEA/D_{20}$	$MOEA/D_{210}$	$MOEA/D_{N_{spec}}$
50	541.6037	566.8921	621.2476	620.8365	486.5366	532.4463	536.6768
75	642.7674	653.4373	714.5162	720.6106	545.0088	587.8820	590.2881
100	673.7599	687.6201	742.6952	738.5506	557.9805	582.8894	587.8565
125	736.4336	753.2498	792.7041	790.5849	739.4365	763.5474	780.1217
150	772.5983	787.7714	857.3251	851.2659	612.6652	640.4161	638.2581
175	783.4265	803.4243	838.2117	835.0432	636.6017	653.1362	657.8071
200	773.0628	788.0596	854.1746	848.2873	614.7756	637.8039	642.2385
225	876.5625	892.6258	948.5553	934.2622	651.9642	663.5777	693.1615
250	905.2262	925.0562	962.4042	971.3175	775.4714	794.873	840.8401

From the results presented in Table 2 it is clear that the Sim-EA algorithm produces the best results mainly with the "rank" migration strategy. The "uniform" migration strategy usually gives a little worse results, but for two datasets it outperformed the "rank" strategy. Graphs presented in Figure 3 show that the median value of the objectives obtained by the Sim-EA algorithm increases approximately linearly with the instance size N_v. In the case of the MOEA/D algorithm there is a sharp increase in the obtained median values for $N_v = 125$ and a decrease for the range $N_v \in [150, 200]$.

In order to verify the statistical significance of the results the Wilcoxon rank test [22] was performed. This particular test was used because it does not assume the normality of the distributions which may be hard to ensure in practical

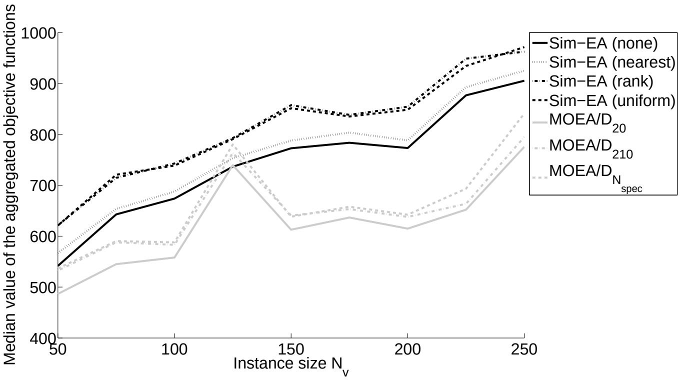

Fig. 3. Median values of the aggregated objective functions obtained in the experiments.

applications. Also, this test was recommended in a recent survey [10] as one of the methods suitable for statistical comparison of results produced by metaheuristic optimization algorithms. Table 3 presents, for each instance size N_v, the best and the second-best algorithm and the p-value obtained in the statistical test with the null hypothesis that the median values of the aggregated objective functions are the same for both algorithms. The p-values that correspond to the significance of 95% or better (i.e. p-value ≤ 0.05) are underlined. Obviously, such low p-values indicate, that the algorithm which produced higher median in the experiments is significantly better than the other one.

Table 3. Best performing algorithms and the results of the statistical verification.

N_v	Best algorithm	Second-best algorithm	p-value
50	Sim-EA (rank)	Sim-EA (uniform)	$4.0938 \cdot 10^{-1}$
75	Sim-EA (uniform)	Sim-EA (rank)	$3.5698 \cdot 10^{-1}$
100	Sim-EA (rank)	Sim-EA (uniform)	$\underline{2.5766 \cdot 10^{-6}}$
125	Sim-EA (rank)	Sim-EA (uniform)	$\underline{2.3406 \cdot 10^{-7}}$
150	Sim-EA (rank)	Sim-EA (uniform)	$\underline{8.1925 \cdot 10^{-14}}$
175	Sim-EA (rank)	Sim-EA (uniform)	$3.9639 \cdot 10^{-1}$
200	Sim-EA (rank)	Sim-EA (uniform)	$\underline{1.4020 \cdot 10^{-2}}$
225	Sim-EA (rank)	Sim-EA (uniform)	$\underline{1.9732 \cdot 10^{-5}}$
250	Sim-EA (uniform)	Sim-EA (rank)	$\underline{1.4062 \cdot 10^{-2}}$

In the case of $N_v = 100$, 125, 150, 200 and 225 the statistical test confirmed that the difference in favour of the "rank" strategy is significant at the significance level 0.05. In the case of $N_v = 50$ and 175 the significance was not confirmed. Obviously, for $N_v = 75$ and 250 the differences are in favour of the "uniform" strategy. These differences were shown to be insignificant in the case of $N_v = 75$ and significant in the case of $N_v = 250$.

5 Conclusion

In this paper the multipopulation algorithm Sim-EA was applied to the multi-objective firefighter problem. In the experiments it was observed that the multipopulation approach can utilize computational resources more effectively than a traditional decomposition-based method. Migration significantly improves the results obtained by the Sim-EA algorithm. In all cases when any form of migration was used the Sim-EA algorithm performed better than in the case when no migration was used. The best performing migration scheme is the "rank" scheme in which the source population for migration is randomly chosen with probability proportional to the rank based on similarity to the destination population with respect to the search directions associated with these subpopulations. Another well performing migration scheme is the "uniform" scheme in which the source population is randomly chosen with uniformly distributed probability. For two datasets it even outperformed the "rank" scheme. In the remaining cases the results are in favour of the "rank" scheme.

Further work on the presented topic is currently underway and it is focused on applying the approach described in this paper to dynamic optimization of a non-deterministic version of the firefighter problem.

References

1. Removed due to the anonymization requirement
2. Removed due to the anonymization requirement
3. Anderson, P.G., Ashlock, D.: Advances in ordered greed. In: Dagli, C.H. (ed.) Intelligent Engineering Systems through Artificial Neural Networks, Proceedings of ANNIE 2004 International Conference. pp. 223–228. ASME Press, New York (2004)
4. Bierwirth, C., Mattfeld, D.C., Kopfer, H.: On permutation representations for scheduling problems. In: In 4th PPSN. pp. 310–318. Springer-Verlag (1996)
5. Blanton, Jr., J.L., Wainwright, R.L.: Multiple vehicle routing with time and capacity constraints using genetic algorithms. In: Proceedings of the 5th International Conference on Genetic Algorithms. pp. 452–459. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
6. Blum, C., Blesa, M., Garca-Martnez, C., Rodrguez, F., Lozano, M.: The firefighter problem: Application of hybrid ant colony optimization algorithms. In: Blum, C., Ochoa, G. (eds.) Evolutionary Computation in Combinatorial Optimisation, Lecture Notes in Computer Science, vol. 8600, pp. 218–229. Springer Berlin Heidelberg (2014), `http://dx.doi.org/10.1007/978-3-662-44320-0_19`
7. Cicirello, V.A., Smith, S.F.: Modeling GA performance for control parameter optimization. In: Whitley, L. (ed.) GECCO-2000: Proceedings of the Genetic and Evolutionary Computation Conference: A Joint Meeting of the Ninth International Conference on Genetic Algorithms (ICGA-2000) and the Fifth Annual Genetic Programming Conference (GP-2000), July 10-12, 2000, Las Vagas, Nevada, pp. 235–242. Morgan Kaufmann Publishers (2000)
8. Cicirello, V.A.: 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. pp. 1125–1132. ACM, New York, NY, USA (2006)

9. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6, 182–197 (2002)
10. Derrac, J., Garca, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation 1(1), 3–18 (2011)
11. Develin, M., Hartke, S.G.: Fire containment in grids of dimension three and higher. Discrete Appl. Math. 155(17), 2257–2268 (2007)
12. Falkenauer, E., Bouffouix, S.: A genetic algorithm for job shop. In: Proceedings of the 1991 IEEE International Conference on Robotics and Automation. pp. 824–829 (1991)
13. Feldheim, O.N., Hod, R.: 3/2 firefighters are not enough. Discrete Applied Mathematics 161(12), 301–306 (2013)
14. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley (1989)
15. Goldberg, D.E., Lingle Jr., R.: Alleles, loci, and the traveling salesman problem. In: Grefenstette, J.J. (ed.) Proceedings of the First International Conference on Genetic Algorithms and Their Applications. pp. 154–159. Lawrence Erlbaum Associates, Publishers (1985)
16. Hartnell, B.: Firefighter! an application of domination. In: 20th Conference on Numerical Mathematics and Computing (1995)
17. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation 13(2), 284–302 (2009)
18. Miller, B.L., Goldberg, D.E.: Genetic algorithms, tournament selection, and the effects of noise. Complex Systems 9, 193–212 (1995)
19. Mumford, C.L.: New order-based crossovers for the graph coloring problem. In: Proceedings of the 9th International Conference on Parallel Problem Solving from Nature. pp. 880–889. Springer-Verlag, Berlin, Heidelberg (2006)
20. Oliver, I.M., Smith, D.J., Holland, J.R.C.: A study of permutation crossover operators on the traveling salesman problem. In: Proceedings of the Second International Conference on Genetic Algorithms on Genetic Algorithms and Their Applications. pp. 224–230. Lawrence Erlbaum Associates Inc., Hillsdale, NJ, USA (1987)
21. Syswerda, G.: Schedule optimization using genetic algorithms. In: Davis, L. (ed.) Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York, NY (1991)
22. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bulletin 1(6), 80–83 (1945)
23. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation 11(6), 712–731 (2007)