Sim-EA: An Evolutionary Algorithm Based on Problem Similarity

Krzysztof Michalak

Department of Information Technologies,
Institute of Business Informatics,
Wroclaw University of Economics, Wroclaw, Poland
krzysztof.michalak@ue.wroc.pl

Abstract. In this paper a new evolutionary algorithm Sim-EA is presented. This algorithm is designed to tackle several instances of an optimization problem at once based on an assumption that it might be beneficial to share information between solutions of similar instances. The Sim-EA algorithm utilizes the concept of multipopulation optimization. Each subpopulation is assigned to solve one of the instances which are similar to each other. Problem instance similarity is expressed numerically and the value representing similarity of any pair of instances is used for controlling specimen migration between subpopulations tackling these two particular instances.

Keywords: multipopulation algorithms, evolutionary optimization, combinatorial optimization, travelling salesman problem

1 Introduction

This paper proposes an evolutionary algorithm Sim-EA which is designed to solve multiple similar instances of an optimization problem by utilizing the idea of a multipopulation evolutionary algorithm.

Probably the most common motivation for employing multipopulation evolutionary algorithms is diversity preservation. Multipopulation evolutionary algorithms are commonly used for such types of problems for which converging to a single optimum is not good enough. This situation is typical to multimodal problems in which there are many equally good or almost equally good solutions with different parameters. It is usually desirable to find many such solutions to allow the decision-maker to choose the best option. Techniques such as species conservation [7], algorithms based on an island model [1] and small-world topologies [5] were applied in the literature to multimodal problems.

Another area in which multipopulation algorithms are often used is multiobjective optimization. Because in multiobjective optimization solutions are evaluated using several different criteria it is not desirable to select only one solution arbitrarily. Instead, an optimization algorithm is usually expected to return an entire Pareto front of nondominated solutions. In [9] parallel approaches to multiobjective optimization are reviewed.

The third type of problems in which multipopulation algorithms are commonly used are dynamic optimization problems. In this type of problems it is undesirable for the population to converge to a single optimum because when the environment changes it is very hard to restore diversity in the population and to start searching for a new optimum. Multipopulation algorithms proposed for this type of problems include forking genetic algorithms (FGAs) [11], Shifting Balance GA (SBGA) [3], the multinational GA (MGA) [12] and Self-Adaptive Differential Evolution algorithm (jDE) [2].

In this paper the multipopulation approach is used for a different purpose: to organize information exchange between simultaneous attempts of solving a set of similar instances on an optimization problem.

The migration scheme influences the information exchange between populations and thus influences the behaviour of the algorithm. In the algorithm proposed in this paper subpopulations are explicitly assigned to different instances of the optimization problem and thus are required to converge to different optima. A similarity measure is used to control the migration of specimens between populations so as to promote information exchange between similar subproblems.

The rest of this paper is structured as follows. In Section 2 the Sim-EA algorithm is described. Section 3 describes the experimental setup and presents the results of the experiments. Section 4 concludes the paper.

2 Algorithm Description

The Sim-EA algorithm proposed in this paper is based on the idea of a multipopulation evolutionary algorithm. The overview of the Sim-EA algorithm is presented in Algorithm 1. The parameters of the algorithm are: N_{gen} - the number of generations, N_{pop} - the size of each subpopulation, N_{prob} - the number of problem instances, N_{imig} - the number of migrated specimens.

In the Sim-EA algorithm subpopulations are assigned to different instances of a given optimization problem. It is assumed that a certain similarity measure can be used to quantitatively describe the similarity of problem instances. Denote the number of problem instances solved simultaneously by N_{prob}. We assume, that a similarity matrix $S_{[N_{prob} \times N_{prob}]}$ is given or can be calculated in the preprocessing phase of the algorithm. For example, if the algorithm solves 20 instances of the Travelling Salesman Problem involving K cities we have 20 cost matrices $C^{(1)}_{[K \times K]}, C^{(2)}_{[K \times K]}, \ldots, C^{(20)}_{[K \times K]}$ that contain distances between the cities (or some other travel cost measure). The similarity between instances i and j, where $i, j \in \{1, \ldots, 20\}$ is simply the similarity of the cost matrices $C^{(i)}_{[K \times K]}$ and $C^{(j)}_{[K \times K]}$. The similarity of such two matrices can be, for example, calculated as $S_{i,j} = -\sum_{p=1}^{K} \sum_{q=1}^{K} (C^{(i)}_{p,q} - C^{(j)}_{p,q})^2$. A similarity matrix is subsequently used to control migration of specimens between populations. Obviously, many migration strategies are possible. Preliminary research shown that one of the best performing ones is to migrate N_{imig} best specimens from one subpopulation which is

Algorithm 1 The overview of the Sim-EA algorithm.

```
IN:
    N_gen - the number of generations
    N_pop - the size of each subpopulation
    N_prob - the number of problem instances
    N_imig - the number of migrated specimens

Calculate the problem instance similarity matrix S_[N_prob × N_prob]
Initialize subpopulations P_1, P_2, ..., P_{N_prob}.
for g = 1, ..., N_gen do
    Apply genetic operators
    for d = 1, ..., N_prob do
        s = argmax_t(S_{d,t})
        P'_d = the N_imig best specimens from P_s
    end for
    for d = 1, ..., N_prob do
        for x ∈ 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 ∪ {b}
        end for
    end for
    Apply genetic operators
    for d = 1, ..., N_prob do
        e = the best specimen in P_d
        P_d = Select(P_d\{e}, N_pop − 1)
        P_d = P_d ∪ {e}
    end for
end for
```

working on the most similar problem instance. In the Sim-EA algorithm the migration to population P_d is performed as follows. The N_{imig} best specimens are selected from the population P_s which solves the most similar problem instance and placed in a set P'_d. After all the sets P'_d, where $d = 1, \ldots, N_{prob}$ are selected the immigrants are merged into respective populations (i.e. each P'_d is merged into the respective P_d). The merging phase is performed using the binary tournament [8] procedure in which each immigrant is compared to the current weakest specimen in the existing population P_d. If the immigrant wins the tournament it replaces the weakest specimen in the population P_d. Genetic operators are applied before and after the migration phase. The aim of the second application of genetic operators is to allow the information from migrated specimens to be incorporated into the target population before the selection step. The selection phase can be performed using any selection procedure such as a roulette wheel selection or a binary tournament. In the proposed algorithm the elitism is used, i.e. the best specimen in each subpopulation P_d is always promoted to the next generation.

3 Experiments and Results

The experiments were performed on a single-objective version of the Travelling Salesman Problem (TSP) [6]. Because in this problem solutions are represented as permutations the Inver-Over genetic operator [10] was used. In the selection phase the binary tournament selection procedure was used. Evolutionary algorithms, especially those dealing with combinatorial optimization are often augmented with local search procedures. In this paper the 2-opt local search [13] was used to improve the quality of the results. Parameters of the Sim-EA algorithm used in the experiments are summarized in Table 1.

In order to verify the assumption that it is beneficial to perform migration based on problem instance similarity, three different strategies were compared: **1-nearest-N-best** (N_{imig} best specimens are migrated from the nearest population), **1-uniform-N-best** (N_{imig} best specimens are migrated from one population which is selected randomly with uniform probability distribution among populations) and **none** (no migration is performed).

Table 1. Parameters of the Sim-EA algorithm used in the experiments.

Parameter name	Value
Number of subproblems (N_{prob})	20
Problem size (the number of cities, K)	50
Number of generations (N_{gen})	200
Population size (N_{pop})	100
Random inverse rate for the inver-over operator (δ_i)	0.02

3.1 Test Problem Definition

A set of $N_{prob} = 20$ instances of the TSP with size $K = 50$ each was prepared as follows. The first cost matrix $C^{(1)}_{[K\times K]}$ was randomly initialized by drawing the elements above the diagonal from the uniform probability distribution $U[0, 100]$. To obtain a symmetric matrix the elements above the diagonal were copied symmetrically below the diagonal of the matrix. Obviously, the elements on the diagonal were all set to 0.

The remaining cost matrices $C^{(2)}_{[K\times K]}, \ldots, C^{(20)}_{[K\times K]}$ were generated iteratively. The $C^{(j)}_{[K\times K]}$ matrix was generated from the $C^{(j-1)}_{[K\times K]}$ by replacing $1/N_{pop}$ (i.e. $1/20 = 5\%$) of the non-diagonal elements by random values drawn from the uniform probability distribution $U[0, 100]$. Symmetry of the cost matrix was preserved by changing both $C^{(j)}_{m,n}$ and $C^{(j)}_{n,m}$ to the same value.

Clearly, this procedure ensures that the cost matrices $C^{(i)}_{[K\times K]}$ and $C^{(j)}_{[K\times K]}$ are more different for larger differences $|i-j|$. Note, that in the procedure described above no special attention was paid to satisfy the triangle inequality $C^{(i)}_{p,q} + C^{(i)}_{q,r} \geq C^{(i)}_{p,r}$. While this inequality holds in all metric spaces this is not a strict requirement for the TSP problem, because in various applications the cost matrix may represent a non-metric quantity, such as ticket costs, travel risks etc..

3.2 Results

During the experiments 30 iterations of the test were performed for each of the three migration strategies. From the 30 runs median values of the objective function of the best specimen obtained in each run were calculated. Table 2 summarizes median values obtained by each of the migration strategies for each of the subproblems. Obviously, lower values are better (lower travel costs). For each subproblem the best of the values obtained using the three strategies is marked in bold in Table 2.

Table 2. Median values of the travel cost obtained by each of the migration strategies for each of the subproblems.

Subproblem	1-nearest N-best	1-uniform N-best	none	Subproblem	1-nearest N-best	1-uniform N-best	none
1	**171.3077**	176.1562	176.0278	11	**219.3591**	221.2456	221.6789
2	**179.3029**	182.5553	183.6177	12	**223.8487**	226.7204	227.2988
3	**190.1753**	193.9355	195.0356	13	211.1367	**210.3828**	214.3309
4	**194.6214**	198.8933	201.1124	14	**212.1491**	214.721	214.4843
5	**186.2072**	192.4362	192.5718	15	**222.9707**	223.0239	224.8621
6	**180.9736**	186.2206	187.4909	16	**228.5528**	230.5679	229.6890
7	**188.1285**	191.1580	192.2368	17	**263.6930**	266.6074	268.6715
8	**193.7582**	194.4849	198.1625	18	**262.3891**	263.0081	266.0693
9	**202.9813**	206.9935	209.6731	19	**257.2304**	258.7912	260.6183
10	**223.4921**	225.8248	227.4927	20	**249.1812**	252.2827	253.0103

With the exception of the subproblem #13 the 1-nearest-N-best migration strategy gave the best results in the tests. The 1-uniform-N-best migration strategy outperformed the algorithm with no migration in the case of all subproblems except #1, #14 and #16. Clearly, there is some advantage in migrating specimens between subproblems even at random, but the migration based on problem instance similarity has the advantage over other tested approaches.

In order to verify the significance of the results statistical testing was performed. Because the normality of distribution of the measured values cannot be guaranteed the Wilcoxon rank test [14] which does not assume normality was used. This test was recommended in a recent review article [4] which analyzed various methods of statistical testing of the results given by metaheuristic methods. Table 3 summarizes the statistical tests. For each of the subproblems the p-values are given for the null hypothesis that the 1-nearest-N-best strategy gives worse results than each of the two remaining strategies (1-uniform-N-best and none). In Table 3 the interpretation of the p-values is also given. The interpretation is "signif." if the median value obtained by the 1-nearest-N-best strategy is lower than the other one and the p-value is ≤ 0.05. If the median value obtained by the 1-nearest-N-best strategy is lower than the other one, but the p-value is larger than 0.05 the interpretation is "insignif.". If the median value obtained by the 1-nearest-N-best strategy is higher than the other one the interpretation is "worse".

Table 3. The p-values for the null hypothesis that the 1-nearest-N-best strategy gives worse results than each of the two remaining strategies obtained in the statistical verification of the experimental results. Low p-values (≤ 0.05) indicate that 1-nearest-N-best strategy is significantly better than the strategy to which it is compared

Sub-problem	vs. none		vs. 1-uniform-N-best		Sub-problem	vs. none		vs. 1-uniform-N-best	
	p-value	interp.	p-value	interp.		p-value	interp.	p-value	interp.
1	0.0001891	signif.	0.00017423	signif.	11	0.0082167	signif.	0.11093	insignif.
2	0.00061564	signif.	0.0054597	signif.	12	0.00096266	signif.	0.0014839	signif.
3	0.00052872	signif.	0.0082167	signif.	13	0.057096	insignif.	0.97539	worse
4	3.1123e-005	signif.	2.163e-005	signif.	14	0.013194	signif.	0.17138	insignif.
5	0.0001057	signif.	0.0003065	signif.	15	0.020671	signif.	0.89364	insignif.
6	6.3391e-006	signif.	8.4661e-006	signif.	16	0.40483	insignif.	0.13059	insignif.
7	1.2381e-005	signif.	0.0064242	signif.	17	0.00048969	signif.	0.031603	signif.
8	0.00066392	signif.	0.036826	signif.	18	0.0019646	signif.	0.4908	insignif.
9	9.3157e-006	signif.	0.00035888	signif.	19	0.0024147	signif.	0.17791	insignif.
10	0.0017088	signif.	0.0046818	signif.	20	0.00083071	signif.	0.0046818	signif.

Clearly, the 1-nearest-N-best strategy significantly outperforms the others in most cases. For six subproblems the obtained results are better than for the 1-uniform-N-best strategy, but without statistical significance. For the subproblem #13 the results obtained using the 1-uniform-N-best strategy are better than the results produced by the 1-nearest-N-best strategy.

The dynamic behaviour of the algorithm is presented in Figure 1. In these figure the median values of the best specimen cost calculated over all 30 runs are presented for subproblems #1, #10 and #20. The subproblem #13 for which the 1-nearest-N-best strategy performed worse than the 1-uniform-N-best strategy is also presented. Note, that the graphs in Figure 1 are plotted against total calculation time. The total calculation time includes the time used for calculating the elements of the similarity matrix $S_{[N_{prob} \times N_{prob}]}$. Obviously, this calculation has to be performed only in the case of the 1-nearest-N-best strategy. The figures present the final half of the evolution because in the first half the values change significantly which makes the figures much less readable.

From the figures it can be seen that even if the calculation time of the similarity matrix $S_{[N_{prob} \times N_{prob}]}$ is taken into account the 1-nearest-N-best strategy outperforms the others. Also, even though for the subproblem #13 the 1-uniform-N-best strategy gives the best results the difference between this strategy and 1-nearest-N-best is very small.

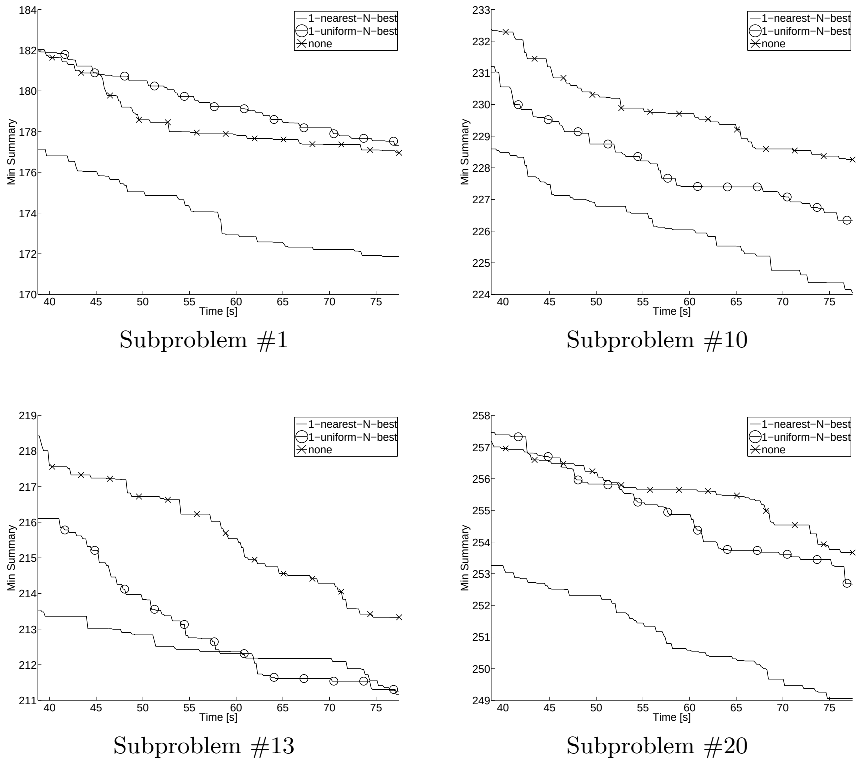

Fig. 1. Median values of the best specimen cost calculated over all 30 runs for subproblems #1, #10, #13 and #20

4 Conclusion

In this paper a new algorithm Sim-EA is proposed. This algorithm uses a multi-population approach to solve several similar instances of an optimization problem simultaneously. A similarity measure is used to determine to what extent are the subproblems similar. This similarity measure is used in the 1-nearest-N-best strategy to select one population which solves the most similar subproblem for the migration of specimens.

The presented algorithm was tested on a set of 20 Travelling Salesman Problem instances with cost matrices that have varying degree of similarity to each other. The experiments show that the 1-nearest-N-best migration strategy which uses the similarity measure defined in this paper is performing better than the 1-uniform-N-best strategy which selects the source for the migration randomly. Without migration the results are even further deteriorated.

Further work may include experiments with updating the similarity measure during the algorithm runtime based on information discovered during the optimization process.

References

1. Bessaou, M., Petrowski, A., Siarry, P.: Island model cooperating with speciation for multimodal optimization. In: Schoenauer, M., Deb, K., Rudolph, G., Yao, X., Lutton, E., Guervs, J.J.M., Schwefel, H.P. (eds.) PPSN. Lecture Notes in Computer Science, vol. 1917, pp. 437–446. Springer (2000)
2. Brest, J., Zamuda, A., Boskovic, B., Maucec, M.S., Zumer, V.: Dynamic optimization using self-adaptive differential evolution. In: IEEE Congress on Evolutionary Computation. pp. 415–422. IEEE (2009)
3. Chen, J., Wineberg, M.: Enhancement of the shifting balance genetic algorithm for highly multimodal problems. In: Proceedings of the 2004 IEEE Congress on Evolutionary Computation. pp. 744–751. IEEE Press, Portland, Oregon (2004)
4. Derrac, J., et al.: 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)
5. Giacobini, M., Preuss, M., Tomassini, M.: Effects of scale-free and small-world topologies on binary coded self-adaptive cea. In: Proceedings of the 6th European Conference on Evolutionary Computation in Combinatorial Optimization. pp. 86–98. EvoCOP'06, Springer-Verlag, Berlin, Heidelberg (2006)
6. Gutin, G., Punnen, A. (eds.): The Traveling Salesman Problem and Its Variations. Combinatorial Optimization, Springer (2007)
7. Li, J.P., Balazs, M.E., Parks, G.T., Clarkson, P.J.: A species conserving genetic algorithm for multimodal function optimization. Evol. Comput. 10(3), 207–234 (September 2002)
8. Miller, B.L., Goldberg, D.E.: Genetic algorithms, tournament selection, and the effects of noise. Complex Systems 9, 193–212 (1995)
9. Talbi, E.G., Mostaghim, S., Okabe, T., Ishibuchi, H., Rudolph, G., Coello Coello, C.A.: Parallel approaches for multiobjective optimization. In: Branke, J., Deb, K., Miettinen, K., Sowiski, R. (eds.) Multiobjective Optimization, Lecture Notes in Computer Science, vol. 5252, pp. 349–372. Springer Berlin Heidelberg (2008)
10. Tao, G., Michalewicz, Z.: Inver-over operator for the tsp. In: Eiben, A.E., Bck, T., Schoenauer, M., Schwefel, H.P. (eds.) Parallel Problem Solving from Nature PPSN V, Lecture Notes in Computer Science, vol. 1498, pp. 803–812. Springer Berlin Heidelberg (1998)
11. Tsutsui, S., Fujimoto, Y., Ghosh, A.: Forking genetic algorithms: Gas with search space division schemes. Evolutionary Computation 5(1), 61–80 (1997)
12. Ursem, R.K.: Multinational GA Optimization Techniques in Dynamic Environments. In: Whitley, D., Goldberg, D., Paz, C.E., Spector, L., Parmee, I., Beyer, H.G. (eds.) Genetic and Evolutionary Computation Conference. pp. 19–26. Morgan Kaufmann (2000)
13. Watson, J., et al.: The traveling salesrep problem, edge assembly crossover, and 2-opt. In: Eiben, A.E., Bck, T., Schoenauer, M., Schwefel, H.P. (eds.) Parallel Problem Solving from Nature PPSN V, Lecture Notes in Computer Science, vol. 1498, pp. 823–832. Springer Berlin Heidelberg (1998)
14. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bulletin 1(6), 80–83 (1945)