This article can be cited as E. Osaba, E. Onieva, F. Dia, R. Carballedo, P. Lopez and A. Perallos, A migration strategy for distributed evolutionary algorithms based on stopping non-promising subpopulations: A case study on routing problems, International Journal of Artificial Intelligence, vol. 13, no. 2, pp. 46-56, 2015.

A migration strategy for distributed evolutionary algorithms based on stopping non-promising subpopulations: A case study on routing problems

Eneko Osaba[1], Enrique Onieva[1], Fernando Diaz[2], Roberto Carballedo[1], Pedro Lopez[1] and Asier Perallos[1]

[1]Deusto Institute of Technology (DeustoTech), University of Deusto
Av. Universidades 24, Bilbao 48007, Spain
{e.osaba, enrique.onieva, roberto.carballedo, p.lopez, perallos}@deusto.es

[2]Telecommunications Department, University of Deusto
Av. Universidades 24, Bilbao 48007, Spain
fernando.diaz@deusto.es

ABSTRACT

Nowadays, parallel genetic algorithms are one of the most used meta-heuristics for solving combinatorial optimization problems. One of the challenges that arise when implementing these kind of algorithms is the communication between subpopulations. This communication, called migration, is a determining factor for a good performance of the algorithm. In this short note, a new approach for the subpopulations communication is presented. This new strategy is called Standstill & Parade. The basis of this new strategy is to stop non-promising subpopulations, in order to focus the search on those that demonstrate more effectiveness. To prove the quality of this approach seven different migration functions are compared. For the experimentation, two different routing problems have been used.

Keywords: Distributed Genetic Algorithm, Genetic Algorithm, Migration Strategy, Routing Problems, Combinatorial Optimization, Traveling Salesman Problem.

2010 Mathematics Subject Classification: 90C27, 90C59, 68T20.
1998 Computing Classification System: I.2.8., G.1.6.

1 Introduction

Today, parallel genetic algorithms (Alba and Troya, 1999) are one the most used techniques for solving combinatorial optimization problems (Christofides, Mingozzi, Toth and Sandi, 1979). In the literature there are many studies on this type of algorithms (Cantú-Paz, 1998; Knysh and Kureichik, 2010; Luque and Alba, 2011). There are several ways to implement parallel genetic algorithms, but Distributed Genetic Algorithm (DGA) is the most common one. A DGA consists of multiple populations that evolve independently, and communicate with each other occasionally.

The communication between subpopulations is one of the most relevant factors when implementing DGAs. This communication typically involves the sharing of individuals. For this reason, it is called migration. Many studies that highlight the importance of migration phase can be found in the literature (Cantú-Paz, 2001; Alba and Troya, 2000).
As a result of this, researchers have proposed diverse migration strategies, each with its own characteristics. In this short note a new migration technique, called Standstill & Parade (S&P), is presented. To prove the quality of this new approach, an experimental study is done, comparing seven different migration functions of the literature. The main novelty of the proposed strategy is its stopping mechanism. Thanks to this mechanism, the technique avoids wasting excessive amounts of time running subpopulations with low capacity of improvement. Additionally, the presented strategy counts with a restart procedure, which provides enough diversity to the subpopulations to continue the search efficiently.
This short note is structured as follows. In Section 2 the proposed approach is described. Section 3 shows the experimentation. This note finishes with the conclusions, in Section 4.

2 Standstill & Parade

The proposed communication strategy is based on two separate concepts. The first one is called *Standstill*. Its purpose is to stop the evolution process of inefficient subpopulations (or demes). The second one, *Parade*, performs the exchange of individuals between demes. In this section, the working way of the S&P is detailed. Furthermore, the pseudo-code of a DGA using S&P can be seen in Algorithm 1.
For the *Standstill* process, each subpopulation has a variable T_i, which measures the progressive improvement of the subpopulation i. A high value of T_i means that the subpopulation fitness is improving over the last generations. On the other hand, a low value of T_i indicates that the search process of the deme is not obtaining better results. T_i is recalculated every generation depending on the search performance. For example, if the overall fitness of a subpopulation i does not improve over the last generation, its T_i decreases. Otherwise, T_i increases. In addition, a new variable (called Thr) that acts as lowerbound for T_i is defined. In this work, the value of Thr is the same for all the demes. If at any point of the execution, the T_i of a subpopulation i reaches Thr, the search process of this subpopulation is stopped. This detention aims to save computational resources consumed by non-promising demes. Thus, the search process focuses on the most promising ones.
When a subpopulation is stopped, the rest of the demes continue the execution until the value T_j of another subpopulation j reaches Thr. At this moment, the *Parade* is performed, and subpopulations i and j combine their individuals. In this first version of the strategy, i and j exchange the 50% of their population. After this combination, i and j restart their T_i and T_j, and the process continues. The whole S&P process is represented in Figure 1. Finally, Algorithm 2 depicts the pseudo-code of the *Parade* procedure.

Algorithm 1: Pseudocode of a DGA with S&P migration strategy

```
initialization and creation of the subpopulations
repeat
    for each subpopulation i do
        if subpopulation i is not stopped then
            Parents selection process
            Crossover phase
            Mutation phase
            Survivor selection process
            if best solution of population i has been improved then
                T_i is restarted
            else
                T_i is decreased
                if T_i has reached Thr and the number of subpopulations stopped is 0 then
                    Standstill(i) (Subpulation i is stopped)
                else
                    (In this case, it means that there is another population j stopped)
                    Parade(i,j) (Subpopulation i and j combine their individuals)
                    T_i and T_j are restarted
                end
            end
        end
    end
until termination criterion reached;
Return the fitness of the best individual found
```

3 Experimentation

To check the quality of the proposed technique, the performance of eight different versions of a basic DGA applied to two well-known routing problems is compared. The problems used are the Traveling Salesman Problem (TSP) (Lawler, Lenstra, Kan and Shmoys, 1985), and the Capacitated Vehicle Routing Problem (CVRP) (Laporte, 1992). It is important to highlight that these problems have been used as benchmarking problems, that is, the main objective of this study is not to find an optimal solution to these problems. In fact, in the literature there are multiple efficient techniques with this objective (Victer Paul,

Algorithm 2: Pseudocode of $Parade$ process with $subpopulation_i$ and $subpopulation_j$

```
jointPopulations = subpopulation_i + subpopulation_j (individuals in jointPopulations are ordered randomly)
Create two empty subpopulations (subpopulation'_i and subpopulation'_j)
repeat
    Add to subpopulation'_i the first individual of jointPopulations
    Add to subpopulation'_j the second individual of jointPopulations
    Remove the first and second individuals of jointPopulations
until jointPopulations is empty;
Return subpopulation'_i and subpopulation'_j
```

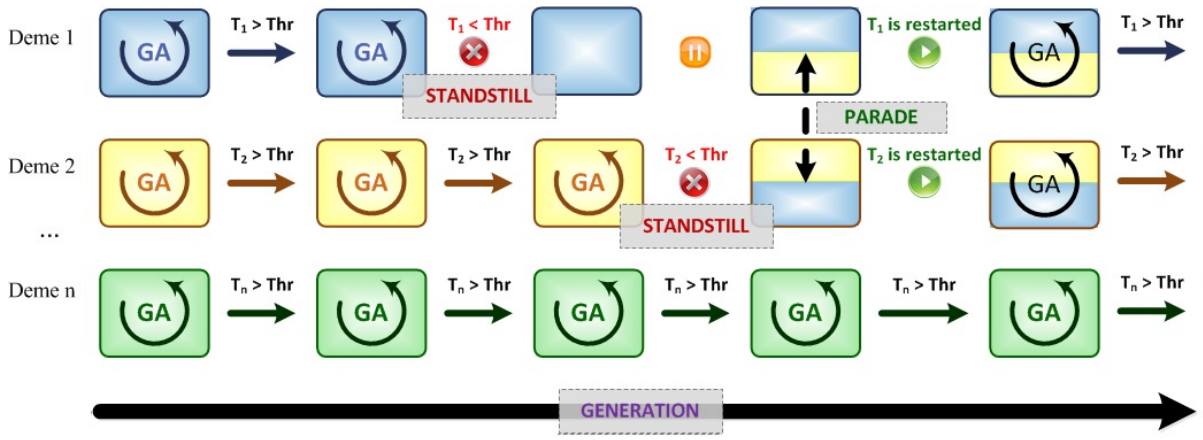

Figure 1: Representation of the Standstill & Parade process

Ramalingam, Baskaran, Dhavachelvan, Vivekanandan and Subramanian, 2014; Ouaarab, Ahiod and Yang, 2014; Golden, Wasil, Kelly and Chao, 1998). Therefore, the objective of using the TSP and CVRP is to compare the performance of the different migration techniques applied to them, and to conclude which one obtains better results.

All the DGAs use the same parameters and functions. In this way, the only difference between them is the migration strategy. For all the DGAs, the whole population is divided into 6 subpopulations of 15 randomly created individuals. All individuals have been encoded using the well-known permutation representation (Toth and Vigo, 2001). As crossover function, the famous Order Crossover has been used (Davis, 1985). Each deme has its own mutation function. The mutation and crossover probabilities are, respectively, 0.2 and 0.8. Binary tournament has been used as parent selection criterion. The survivor function used is the 0.5 elitist - 0.5 random, which means that the half of the survivor population is chosen in elitist way, and the rest randomly. About the ending criteria, the execution of each DGA finishes when there are $n + \sum_{k=1}^{n} k$ generations without improvements in the best solution, where n is the size of the problem. For the S&P, the T_i begins with a value of 100, and the Thr is fixed in 0. If the overall fitness of one subpopulation i does not improve from one generation to the next one, T_i is decreased in 1. On the other hand, if the fitness is improved, T_i is restarted to 100. When T_i =0, subpopulation i is stopped. The 7 migration functions used for the comparison are the following:

- Best-Replace-Worst (BRW) (Cantú-Paz, 2001): In this strategy, every subpopulation i shares its best individual with the following $i + 1$ deme, in a ring topology. This communication happens every generation and the immigrant replaces the worst individual of deme $i + 1$.

- Best-Replace-Random (BRR) (Cantú-Paz, 2001): It works in the same way as BRW. The only difference is that the immigrant replaces one random individual of the subpopulation $i + 1$.

- Randomized Migration Rate (RMR) (Hiroyasu, Miki and Negami, 1999): In this case, each deme i communicates with $i + 1$ at every generation, sharing a random number

of randomly selected individuals. These immigrants join the subpopulation and become part of the genetic process of the next generation.

- MT-GA 2 best version (mt2B) (Cutello, De Michele and Pavone, 2014): In this strategy, every deme i shares its 2 best individuals with a randomly selected subpopulation j. Shared individuals joint the new population.

- MT-GA 1best - 1random version (mt1B1R) (Cutello et al., 2014): It works similarly to the previous one. But in this case deme i shares two individuals, the best one, and a randomly selected one.

- Immigrant Pool Strategy (IPS) (Lopes, Silva, Campelo and Guimarães, 2013): In this approach an immigrant pool is created with the best individual of each deme. Then, at each generation, every subpopulation chooses its immigrant using the Roulette Wheel selection technique. This immigrant joints the deme.

- PROACT (Salto, Luna and Alba, 2013): In this communication strategy each deme has an entropy value $H(g_i)$, in the interval [0,1]. Migration frequency depends on this parameter. When it is close to zero, subpopulation i ask to $i+1$ to send individuals with higher frequency. On the other hand, when it is close to 1.0 the communication is decreased. In this strategy, the selection and replacement strategies are, respectively, sending the best individual, and replacing the worst one.

These seven migration strategies have been chosen based on several factors. BRP, BRR and RMR have been selected because they have been used in many studies along the history, proving that they are very efficient alternatives. On the other hand, PROACT, IPS, mt2B and mt1B1R have been chosen since they are high quality strategies, which have been presented recently. The objective of this study is to demonstrate that S&P can compete with both classic and recent techniques.

In the performed experimentation 37 different problem instances have been used. These instances have been obtained from the TSPLib (Reinelt, 1991) and the VRPWeb[1]. TSP instances have been extracted from different well-known benchmarks, as the Christofides/Eilon benchmark, and Padberg/Rinaldi benchmark. On the other hand, the first 11 CVRP instances belong to the Christofides/Eilon benchmark (Christofides and Eilon, 1969), and the remaining 4 to the Golden et al. large-scale benchmark (Golden et al., 1998). Each experiment is repeated 20 times. In Table 1 the average and standard deviation for each technique are displayed. Besides this, in order to see if the diferences in the outcomes are significant, a Student's t-test has been performed. The t statistic has the following form:

$$t = \frac{\overline{X_1} - \overline{X_2}}{\sqrt{\frac{(n_1-1)SD_1^2+(n_2-1)SD_2^2}{n_1+n_2-2}\frac{n_1+n_2}{n_1 n_2}}}$$

Where $\overline{X_i}$, SD_i and n_i, are the average, standard deviation and number of executions of each technique, being subindex 1 and 2 for S&P and the compared technique, respectively. The

[1] *http://neo.lcc.uma.es/vrp. Last update: January 2013*

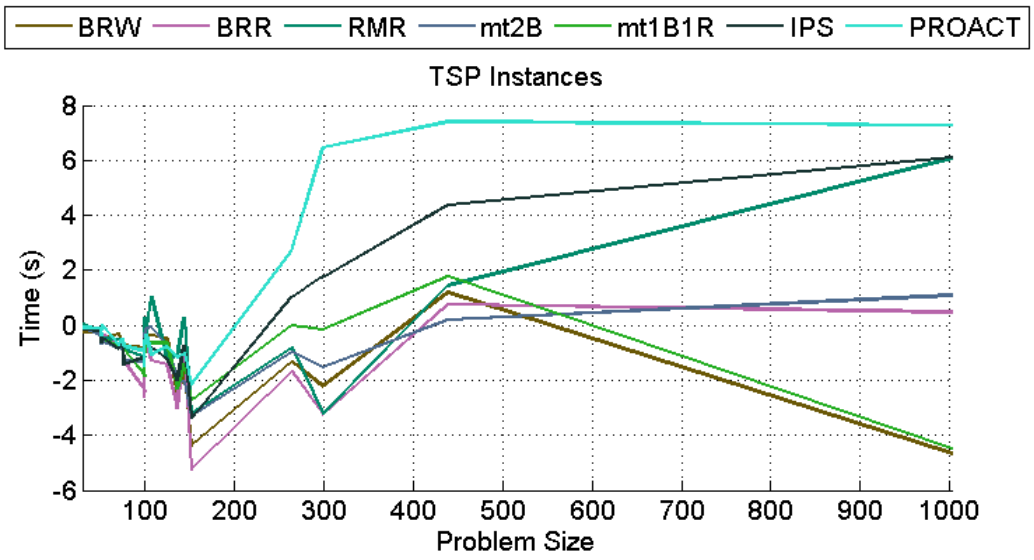

Figure 2: Difference in runtimes (in seconds) between S&P and the other strategies (TSP)

t value can be positive, neutral, or negative. The positive value of t indicates that S&P is significantly better. In the opposite case, S&P obtains worse solutions. If t is neutral, the difference is not significant. The confidence interval has been stated at 95% ($t_{0.05}$ = 2.021). Finally, Figure 2 and Figure 3 compare the runtime of each technique regarding S&P. In these images the differences (in seconds) between the runtimes of S&P and the other strategies are shown. For example, Figure 2 shows that the value obtained by PROACT for an instance of 300 nodes is close to 6. That means that S&P is (in average) 6 seconds faster than PROACT for this instance. On the other hand, if the value is negative, it means that the technique is faster than S&P. For example, RMR is 3 seconds faster than S&P for a 300-node TSP instance.

3.1 Analysis of the results

Viewing these preliminary results some conclusions can be drawn. As can be seen in Table 1, S&P obtains better average values in the 89.18% (33 out of 37) of the instances. In addition, as shown in Table 2, the proposed technique obtains significantly better results in 55.21% (143 out of 259) of the cases. In the remaining 44.79%, the differences in the results are not significant. Besides, S&P never gives significantly worse results. In Table 2, the last row summarizes these outcomes, broken down by technique.

Regarding runtime, Figure 2 shows that, in overall, S&P requires slightly higher runtimes for small TSP instances (less than 200 nodes). On the other hand, for larger instances, S&P needs, generally, less runtime. Anyway, these differences are not remarkable. In fact, for small instances they are located between [-5,+1], and between [-4.5,+7] for large instances. This same analysis can be performed for the CVRP. In this case, for small instances the differences are between [-2.1,+0.5], being slightly worse for S&P. Furthermore, for large instances the differences are located between [-4,+5], although they are slightly better for S&P. Even so, considering the 37 instances used in this study, it should be said that the S&P needs more runtime than the other strategies.

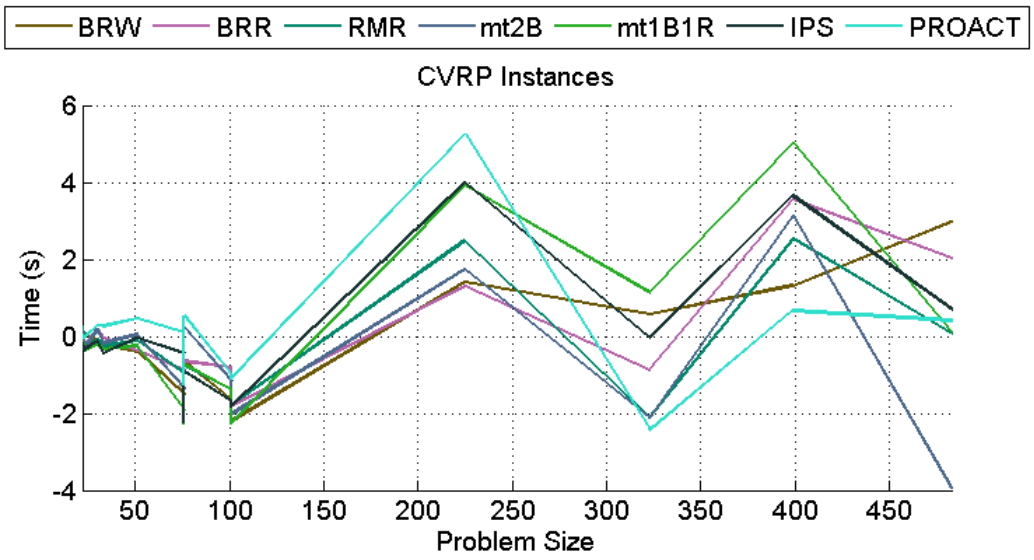

Figure 3: Difference in runtimes (in seconds) between S&P and the other strategies (CVRP)

In conclusion, it can be stated that the objective of this study has been met. With the S&P, communication between demes is performed less frequently than with the other alternatives. This is so because these communications are performed only when they will be beneficial for the search process. This fact gives to S&P a higher exploration capacity, having a greater convergence behavior, and getting better results.

4 Conclusions

In this note some preliminary results of the S&P have been shown. Results has been compared with the ones obtained by seven different migration functions using two well-known routing problems. As future work, it is intended to conduct a more thorough analysis of S&P, performing a convergence behavior study, and applying it to other problems. In addition, a study about the influence of the random parameters is intended to perform. Besides this, the behavior of each subpopulation will be analyzed, calculating the impact of stopping inefficient demes. All this with the intention of improving the performance of the proposed strategy. Furthermore, an interesting future work could be the comparison of the performance of a DGA with the S&P with the one obtained by other different optimization techniques, such as particle swarm optimization (El-Hefnawy, 2014), gravitational search algorithm (David, Precup, Petriu, Rădac and Preitl, 2013), or some other hybrid techniques (Valdez, Melin and Castillo, 2011; Zăvoianu, Bramerdorfer, Lughofer, Silber, Amrhein and Klement, 2013).

Acknowledgement

This work was supported by the Spanish Ministry of Economy and Competitiveness through the *iLogisTICs: Sistema telematico para la mejora de la logistica en el transporte de mercancias a traves de servicios avanzados de planificacion y trazabilidad* project (TEC2013-45585-C2-2-R).

Instance		S&P	BRW	BRR	RMR	mt2B	mt1B1R	IPS	PROACT
Oliver30	420	**421.8** (±2.2)	424.4 (±6.4)	425.2 (±6.3)	423.8 (±5.9)	423.4 (±5.8)	424.7 (±6.6)	426.8 (±8.4)	424.9 (±6.3)
Eilon50	425	**435.5** (±5.4)	439.2 (±7.6)	441.3 (±6.9)	439.4 (±7.1)	440.8 (±4.6)	437.2 (±6.1)	438.3 (±6.1)	437.9 (±5.4)
Eil51	426	**436.1** (±4.2)	440.6 (±6.2)	440.4 (±6.6)	441.1 (±6.0)	440.6 (±6.4)	440.9 (±5.7)	438.8 (±6.0)	440.7 (±5.8)
Berlin52	7542	**7817.3** (±239.3)	7952.7 (±202.6)	7939.8 (±263.0)	7846.9 (±264.9)	7864.6 (±186.5)	7899.0 (±286.2)	7924.6 (±224.6)	7939.6 (±248.2)
St70	675	**697.3** (±14.0)	702.0 (±15.4)	707.1 (±16.9)	700.8 (±11.5)	705.2 (±16.8)	702.3 (±20.5)	707.0 (±17.5)	701.4 (±13.3)
Eilon75	535	561.3 (±9.0)	562.4 (±9.2)	566.7 (±6.6)	566.4 (±8.2)	**559.8** (±7.8)	560.6 (±9.0)	563.0 (±8.4)	565.0 (±8.9)
Eil76	538	**563.4** (±8.5)	566.4 (±9.4)	567.2 (±10.6)	566.5 (±8.1)	566.0 (±7.6)	567.5 (±10.1)	567.9 (±8.9)	565.4 (±11.9)
KroA100	21282	**22019.6** (±465.9)	22538.7 (±600.8)	22478.8 (±725.6)	22521.9 (±620.8)	22406.2 (±680.0)	22151.7 (±501.3)	22346.3 (±553.4)	22091.1 (±677.2)
KroB100	22140	**22962.7** (±407.9)	23455.7 (±365.8)	23353.1 (±492.2)	23112.6 (±416.2)	23257.4 (±431.5)	23171.5 (±438.9)	23069.7 (±503.3)	23106.0 (±450.3)
KroC100	20749	**21493.3** (±407.4)	22216.7 (±656.7)	22001.7 (±480.1)	21728.5 (±438.2)	21844.7 (±599.4)	21959.8 (±635.6)	22007.0 (±670.5)	21810.5 (±497.5)
KroD100	21294	**22232.0** (±465.4)	22521.2 (±450.1)	22491.5 (±453.3)	22572.6 (±453.2)	22486.8 (±412.4)	22420.9 (±563.5)	22652.2 (±537.7)	22472.5 (±372.6)
KroE100	22068	22907.4 (±345.0)	**22841.5** (±415.8)	23228.4 (±404.4)	23028.3 (±413.2)	23026.8 (±348.9)	23215.2 (±476.7)	23286.2 (±550.2)	22903.1 (±403.9)
Eil101	629	**658.1** (±7.8)	665.8 (±8.9)	666.6 (±11.6)	668.6 (±6.4)	667.2 (±7.3)	668.8 (±8.2)	665.6 (±10.4)	664.0 (±7.0)
Pr107	44303	**45758.6** (±853.7)	46863.4 (±1158.8)	47199.7 (±997.6)	46502.0 (±1119.3)	46031.4 (±1422.0)	45947.2 (±622.5)	46220.6 (±1405.9)	46190.0 (±1082.1)
Pr124	59030	**60237.5** (±640.4)	60631.8 (±686.6)	61760.5 (±1450.9)	60631.8 (±1145.2)	60497.6 (±1100.1)	60476.5 (±931.2)	60585.5 (±1183.8)	60863.3 (±1172.4)
Pr136	96772	**101592.3** (±1681.5)	102591.8 (±1520.9)	104164.7 (±2351.4)	102112.0 (±2693.6)	101731.3 (±1661.1)	101735.6 (±1778.1)	102512.0 (±1752.2)	101864.7 (±1595.7)
Pr144	58537	**60059.8** (±1088.5)	60938.2 (±1912.9)	60731.2 (±2256.5)	60466.5 (±1521.2)	60846.6 (±1991.2)	60808.0 (±1668.4)	60866.1 (±1510.9)	62520.5 (±1650.3)
Pr152	73682	**75979.3** (±910.1)	76416.0 (±1254.6)	76639.7 (±1758.9)	76300.3 (±1807.1)	76259.4 (±1499.9)	76214.2 (±1370.1)	76678.6 (±1229.5)	76352.6 (±1191.8)
pr264	49135	**52084.8** (±2002.9)	54041.1 (±1318.6)	54668.7 (±1492.5)	53783.9 (±1333.2)	53779.7 (±812.4)	53841.1 (±1269.9)	53795.0 (±1165.4)	53188.2 (±854.1)
pr299	48191	**50740.9** (±1144.1)	51786.1 (±687.7)	52996.4 (±1158.5)	52273.3 (±927.3)	52022.8 (±1370.2)	52491.4 (±705.3)	52523.6 (±1090.6)	52644.7 (±1032.4)
pr439	107217	**115250.0** (±3825.1)	116797.3 (±1877.9)	119862.8 (±2213.9)	116240.3 (±3064.9)	116412.3 (±1514.6)	117516.7 (±3234.6)	116465.4 (±1801.1)	116911.0 (± 2278.1)
pr1002	259047	**279413.5** (±2130.9)	282205.1 (±3803.8)	290362.6 (±6190.2)	282952.9 (±3664.5)	284302.5 (±3063.8)	284074.6 (±2702.0)	280928.3 (±2882.6)	284086.3 (±2353.7)
En22k4	375	**390.0** (±11.6)	406.7 (±18.8)	394.9 (±12.4)	405.0 (±30.4)	403.4 (±22.5)	403.8 (±18.4)	403.1 (±20.4)	396.1 (±16.4)
En23k3	569	**585.5** (±21.2)	597.3 (±27.7)	602.3 (±29.1)	596.7 (±26.3)	596.4 (±25.1)	593.7 (±24.2)	606.8 (±27.9)	590.8 (±20.7)
En30k3	534	**545.2** (±17.0)	571.0 (±48.2)	572.2 (±74.9)	588.5 (±58.9)	569.9 (±54.5)	570.3 (±48.2)	574.7 (±54.1)	564.0 (±36.8)
En33k4	835	**890.7** (±24.2)	918.6 (±28.6)	906.3 (±23.4)	916.9 (±20.6)	916.4 (±23.1)	915.7 (±22.6)	906.2 (±26.6)	910.2 (±27.1)
En51k5	521	**597.1** (±32.7)	632.9 (±54.1)	630.4 (±41.2)	636.4 (±52.2)	645.2 (±53.0)	646.7 (±34.2)	654.8 (±54.7)	612.6 (±39.8)
En76k7	682	**779.1** (±33.0)	795.1 (±36.9)	831.1 (±37.0)	796.1 (±70.7)	828.3 (±41.4)	817.1 (±31.9)	784.9 (±35.4)	802.9 (±57.8)
En76k8	735	**820.2** (±28.8)	875.0 (±54.3)	881.9 (±53.8)	899.6 (±39.9)	890.3 (±42.3)	874.6 (±46.6)	874.6 (±46.6)	840.3 (±31.3)
En76k10	830	**936.1** (±20.1)	972.7 (±45.2)	997.9 (±58.4)	998.3 (±56.4)	990.7 (±53.9)	1030.5 (±74.7)	979.1 (±54.0)	976.6 (±38.0)
En76k14	1021	1151.8 (±38.5)	1185.3 (±54.0)	1164.0 (±32.1)	1213.5 (±53.7)	1187.4 (±27.9)	1208.0 (±52.8)	1187.5 (±45.3)	**1150.2** (±41.5)
En101k8	815	**952.0** (±39.7)	994.2 (±52.3)	1012.4 (±53.5)	1010.0 (±83.6)	1004.8 (±73.3)	1017.0 (±71.9)	993.7 (±59.3)	979.8 (±24.9)
En101k14	1071	**1250.0** (±38.7)	1273.0 (±38.1)	1267.0 (±69.2)	1309.0 (±49.5)	1307.3 (±68.1)	1271.1 (±48.8)	1270.0 (±60.4)	1267.1 (±58.8)
kelly09(255)	587.09	**738.7** (±23.1)	748.6 (±28.4)	769.6 (±71.2)	767.2 (±24.5)	763.6 (±12.8)	771.6 (±33.1)	750.3 (±21.0)	750.9 (±19.8)
kelly10(323)	746.56	**972.4** (±22.2)	993.2 (±27.3)	1008.1 (±32.3)	996.2 (±36.4)	1003.0 (±27.9)	1007.8 (±17.5)	975.7 (±23.9)	989.3 (±20.3)
kelly11(399)	932.68	**1246.8** (±25.7)	1261.4 (±21.6)	1309.3 (±39.1)	1287.5 (±44.0)	1286.3 (±36.4)	1295.6 (±39.5)	1256.9 (±26.7)	1280.4 (±50.5)
kelly12(483)	1137.18	1397.6 (±35.8)	1405.5 (±23.8)	1419.2 (±43.7)	1402.3 (±40.4)	**1386.5** (±30.8)	1406.4 (±53.3)	1457.5 (±53.2)	1421.1 (±81.1)

Table 1: Results of the 8 different versions of the DGA for TSP and CVRP

Tests for the Traveling Salesman Problem							
Instance	BRW	BRR	RMR	mt2B	mt1B1R	IPS	PROACT
Oliver30	*	+	*	*	*	+	+
Eilon50	*	+	*	+	*	*	*
Eil51	+	+	+	+	+	*	+
Berlin52	*	*	*	*	*	*	*
St70	*	+	*	*	*	*	*
Eilon75	*	+	*	*	*	*	*
Eil76	*	*	*	*	*	*	*
KroA100	+	+	+	+	*	*	*
KroB100	+	+	*	+	*	*	*
KroC100	+	+	*	+	+	+	+
KroD100	+	*	+	*	*	+	*
KroE100	*	+	*	*	+	+	*
Eil101	+	+	+	+	+	+	+
Pr107	+	+	+	*	*	*	*
Pr124	*	+	*	*	*	*	+
Pr136	*	+	*	*	*	*	*
Pr144	*	*	*	*	*	*	+
Pr152	*	*	*	*	*	+	*
pr264	+	+	+	+	+	+	+
pr299	+	+	+	+	+	+	+
pr499	*	*	*	*	*	*	*
pr1002	+	+	+	+	+	*	+
Summary	+ * -	+ * -	+ * -	+ * -	+ * -	+ * -	+ * -
All inst.	10 12 0	16 6 0	8 14 0	9 13 0	7 15 0	8 14 0	9 13 0

Tests for the Capacitated Vehicle Routing Problem							
Instance	BRW	BRR	RMR	mt2B	mt1B1R	IPS	PROACT
En22k4	+	*	+	+	+	+	*
En23k3	*	+	*	*	+	+	*
En30k3	+	*	+	*	+	+	+
En33k4	+	+	+	+	+	*	+
En51k5	+	+	+	+	+	+	*
En76k7	*	+	*	+	+	*	*
En76k8	+	+	+	+	+	+	+
En76k10	+	+	+	+	+	+	+
En76k14	+	*	+	+	+	+	*
En101k8	+	+	+	+	+	+	+
En101k14	*	*	+	+	*	*	*
kelly09	*	*	+	+	+	+	+
kelly10	+	+	+	+	+	*	+
kelly11	*	+	+	+	+	+	+
kelly12	*	*	*	*	*	+	+
Summary	+ * -	+ * -	+ * -	+ * -	+ * -	+ * -	+ * -
All inst.	9 6 0	9 6 0	12 3 0	12 3 0	13 2 0	11 4 0	9 6 0

Table 2: Student's t test for TSP and CVRP. Last row summarizes the results of each confrontation, for every technique, and for both problems

References

Alba, E. and Troya, J. M. 1999. A survey of parallel distributed genetic algorithms, *Complexity* **4**(4): 31–52.

Alba, E. and Troya, J. M. 2000. Influence of the migration policy in parallel distributed gas with structured and panmictic populations, *Applied Intelligence* **12**(3): 163–181.

Cantú-Paz, E. 1998. A survey of parallel genetic algorithms, *Calculateurs Paralleles, Reseaux et Systems Repartis* **10**(2): 141–171.

Cantú-Paz, E. 2001. Migration policies, selection pressure, and parallel evolutionary algorithms, *Journal of Heuristics* **7**(4): 311–334.

Christofides, N. and Eilon, S. 1969. An algorithm for the vehicle-dispatching problem, *Operational Research Quarterly* pp. 309–318.

Christofides, N., Mingozzi, A., Toth, P. and Sandi, C. 1979. *Combinatorial Optimization*, John Willey, Chichester.

Cutello, V., De Michele, A. G. and Pavone, M. 2014. Escaping local optima via parallelization and migration, *Nature Inspired Cooperative Strategies for Optimization*, Springer, pp. 141–152.

David, R.-C., Precup, R.-E., Petriu, E. M., Rădac, M.-B. and Preitl, S. 2013. Gravitational search algorithm-based design of fuzzy control systems with a reduced parametric sensitivity, *Information Sciences* **247**: 154–173.

Davis, L. 1985. Applying adaptive algorithms to epistatic domains, *Proceedings of the International Joint Conference on Artificial Intelligence*, Vol. 1, pp. 161–163.

El-Hefnawy, N. 2014. Solving bi-level problems using modified particle swarm optimization algorithm, *International Journal of Artificial Intelligence* **12**(2): 88–101.

Golden, B. L., Wasil, E. A., Kelly, J. P. and Chao, I.-M. 1998. The impact of metaheuristics on solving the vehicle routing problem: algorithms, problem sets, and computational results, *Fleet Management and Logistics*, Kluwer, Boston, pp. 33–56.

Hiroyasu, T., Miki, M. and Negami, M. 1999. Distributed genetic algorithms with randomized migration rate, *Proceeding of the IEEE International Conference on Systems, Man, and Cybernetics*, Vol. 1, pp. 689–694.

Knysh, D. and Kureichik, V. 2010. Parallel genetic algorithms: a survey and problem state of the art, *Journal of Computer and Systems Sciences International* **49**(4): 579–589.

Laporte, G. 1992. The vehicle routing problem: An overview of exact and approximate algorithms, *European Journal of Operational Research* **59**(3): 345–358.

Lawler, E. L., Lenstra, J. K., Kan, A. R. and Shmoys, D. B. 1985. *The traveling salesman problem: a guided tour of combinatorial optimization*, Vol. 3, Wiley Chichester.

Lopes, R. A., Silva, R. C. P., Campelo, F. and Guimarães, F. G. 2013. Dynamic selection of migration flows in island model differential evolution, *Proceedings of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference Companion*, ACM, pp. 173–174.

Luque, G. and Alba, E. 2011. *Parallel Genetic Algorithms: Theory and Real World Applications*, Vol. 367, Springer.

Ouaarab, A., Ahiod, B. and Yang, X.-S. 2014. Improved and discrete cuckoo search for solving the travelling salesman problem, *Cuckoo Search and Firefly Algorithm*, Springer, pp. 63–84.

Reinelt, G. 1991. Tspliba traveling salesman problem library, *ORSA Journal on Computing* **3**(4): 376–384.

Salto, C., Luna, F. and Alba, E. 2013. Enhancing distributed eas using proactivity, *Proceedings of the Fifteenth Annual Conference on Genetic and Evolutionary Computation Conference Companion*, ACM, pp. 1747–1748.

Toth, P. and Vigo, D. 2001. *The vehicle routing problem*, SIAM, Philadelphia Monograph on Discrete Mathematics and Applications.

Valdez, F., Melin, P. and Castillo, O. 2011. An improved evolutionary method with fuzzy logic for combining particle swarm optimization and genetic algorithms, *Applied Soft Computing* **11**(2): 2625–2632.

Victer Paul, P., Ramalingam, A., Baskaran, R., Dhavachelvan, P., Vivekanandan, K. and Subramanian, R. 2014. A new population seeding technique for permutation-coded genetic algorithm: Service transfer approach, *Journal of Computational Science* **5**(2): 277–297.

Zăvoianu, A.-C., Bramerdorfer, G., Lughofer, E., Silber, S., Amrhein, W. and Klement, E. P. 2013. Hybridization of multi-objective evolutionary algorithms and artificial neural networks for optimizing the performance of electrical drives, *Engineering Applications of Artificial Intelligence* **26**(8): 1781–1794.