Dynamically Tuning the Population Size in Particle Swarm Optimization

André L.V. Coelho
MIA-CCT-Unifor
Washington Soares Av., 1321/J30
60811-905 Fortaleza, CE - Brazil
+55-85-34773268
acoelho@unifor.br

Daniel G. de Oliveira
MIA-CCT-Unifor
Washington Soares Av., 1321/J30
60811-905 Fortaleza, CE - Brazil
+55-85-34773268
goncalvesmail@yahoo.com.br

ABSTRACT
In this paper, we investigate the benefits of dynamically varying the population size in the Particle Swarm Optimization (PSO) model. For this purpose, two well-known population resizing techniques, originally developed for Genetic Algorithms (GAs), were adapted to the PSO context, giving birth to the APPSO and PRoFIPSO variants. Contrary to some previous work that has indicated that the PSO model is not sensitive to the population dimension, the simulation results we have obtained over some benchmark numerical optimization problems suggest that the dynamic variation of the number of particles may be instrumental for bringing about performance improvements in long-term runs, mainly when considering high-dimensional problem instances. In general, the novel PSO variants have compared more favorably to their GA counterparts in targeting the optimal solutions. However, regarding PRoFIPSO specifically, the price to be paid in terms of resources used to reach the optimum point is as a rule very high.

Categories and Subject Descriptors
I.2.8 [**Computing Methodologies**]: Problem Solving, Control Methods, and Search – *heuristic methods*.

General Terms
Algorithms, Experimentation, Performance.

Keywords
Parameter control, Population sizing, Numerical optimization.

1. INTRODUCTION
It is quite consensual nowadays that the population dimension is one of the main aspects that affect the robustness and computational efficiency of any Evolutionary Algorithm (EA). Traditionally, such control parameter is set in advance to a specified value and remains constant through the entire run of the search. However, having to specify the initial value of this parameter is usually a problematic task, since it has been demonstrated, both theoretically and empirically, that the optimal population size is something that depends on the difficulty of the problem under consideration [9][14].

As finding an appropriate population size for each problem is almost impossible, one solution could be working with a variable population size during the evolution. Indeed, some studies have supported this idea, advocating that, at different stages of a single run, different population sizes might be optimal. Based on these observations and motivated by the fact that in natural environments population sizes of species tend to vary continuously and then stabilize around appropriate values [7], in the last years researchers have proposed various schemes that try to learn a good population size during the EA run itself [10]. Recently, Eiben et al. [6] conducted a comparative experimental study applying some of these adaptive population sizing methods to a set of instances of a multimodal problem generator. Amongst them fit the Genetic Algorithm with Adaptive Population Size (APGA) model [2], which is a variant of the Genetic Algorithm with Variable Population Size (GAVaPS) scheme [1], and the Population Resizing on Fitness Improvement Genetic Algorithm (PRoFIGA) model [6].

Taking into account that the dynamic tuning of the population dimension may yield considerable gains to EAs, we have decided to investigate the potentials of applying this concept to other related stochastic population-based metaheuristics. In this paper, we focus on the Particle Swarm Optimization (PSO) model [8], which is motivated from the simulation of social-psychological behavior. Due to its simplicity, flexibility and promising performance on nonlinear function optimization, PSO has received much attention recently. Each potential solution vector in PSO is called a particle, which flies over the search space with a velocity that is dynamically adjusted according to the experiences of that particle and its companions. One distinctive property of PSO is that the same particles are kept as members of the population throughout the course of the run, as no selection operation is performed.

Some previous work [13] has indicated that PSO is not qualitatively sensitive to the prior calibration of the population size based on the shapes of the convergence curves displayed by the algorithm for different values of this parameter. For instance, by employing Clerc's constricted PSO model [4] to optimize five well-known benchmark functions, Carlisle and Dozier [3] varied the population dimension from five to 200 and then deduced that 30 appears to be a good choice for general use. It is worth observing that this conclusion was reached by considering only static values for this parameter and low or moderately-dimensioned (≤ 30) problem instances.

In this paper, we investigate the benefits of dynamically varying the number of particles in constricted PSO via the adaptation of two of the aforementioned EA population resizing techniques, giving birth to the APPSO and PRoFIPSO variants. After conducting several comparative experiments over benchmark numerical optimization problems, some of which are reported in this paper, it is possible for us to assert that the dynamic variation of the number of particles may be instrumental for bringing about performance improvements to PSO in long-term runs (i.e. when there is no limitation on the number of fitness evaluations available) mainly when considering high-dimensional problem instances. On average, in terms of effectiveness (i.e. reaching more closely to the known optimum solutions), the novel PSO variants have compared more favorably to their genetic algorithm (GA) counterparts. Besides, in most of the problem instances, both PRoFIPSO and APPSO could prevail significantly over the constricted PSO model, with APPSO showing a more consistent profile across all problems. However, the price to be paid in terms of efficiency (i.e. number of samplings necessary to locate the optimum points for the first time) was usually very elevated for the PRoFIPSO model, jeopardizing its application in those cases where the fitness evaluation process is expensive.

In the following, we outline the main concepts behind the APGA and PRoFIGA population-size varying schemes, convey a brief overview of the constricted PSO model (also referred to here as standard), give details on how the APGA and PRoFIGA schemes were adapted to the PSO context, and report on some of the simulation experiments we have conducted so far. Finally, the last section concludes the paper and brings remarks on future work.

2. BACKGROUND

The following subsections briefly describe the APGA, PRoFIGA, and standard PSO models, respectively.

2.1 APGA

Following GAVaPS, in APGA, the population size is made adaptive by controlling the birth and death of individuals at each generation of the evolutionary process. Therefore, two novel parameters become associated with each individual: its age and total lifetime. The latter defines the number of generations in which the individual is allowed to remain alive according to its fitness (limited by minimum, *MinLT*, and maximum, *MaxLT*, values), while the first is increased from null up at each iteration. When an individual's age exceeds its total lifetime, it is removed from the population. Basically, there are three possible strategies to assign lifetime values, viz. proportional, linear, and bilinear, the latter being usually regarded as the most consistent one [1][12].

One of the remarkable differences between GAVaPS and APGA is that, in addition to the selection pressure obtained indirectly by the lifetime mechanism, APGA also makes use of an explicit selection operator for choosing individuals to reproduce. Besides, APGA is a steady-state GA, meaning that the reproduction rate has a fixed value (i.e. two individuals) and the best individual in the population does not get older. By this means, APGA prevents the population from growing out of control as it usually happens with GAVaPS [6]. However, as pointed out by Fernandes and Rosa [7], the decision to employ such a low reproduction rate may introduce a "double-sword" effect, turning to be too much restrictive for allowing the population size to freely vary in accord with the necessities of the different stages of the evolution process. In the same vein, Lobo and Lima [11] have recently demonstrated that, after *MaxLT* iterations of the APGA algorithm, the population size is expected to stabilize in a level comparable to the value of *MaxLT*, providing no further source of adaptation at all. These facts are worth being mentioned here beforehand, as they seem to be not applicable to the PSO context (at least in the way we have configured APPSO—see Section 3), as the APPSO scheme could be much better than APGA in the great majority of cases analyzed in Section 4.

2.2 PRoFIGA

In PRoFIGA, the variation of the population dimension is done deterministically and is based solely on the improvement of the best fitness in the population. This scheme intends to balance exploration and exploitation by growing the population in earlier stages and gradually decreasing it in later stages of the search. Whenever the individuals get stuck in local optima, then the process pass through another growing phase, thus increasing diversity and escaping from local optima. This behavior is performed by following three simple rules [6]: (i) if the best fitness in the population increases, the population size will be increased proportionally to the improvement and the number of evaluations left until the maximum allowed; (ii) if there is no improvement during the last V number of evaluations, the population size will also be increased (the amount can be set a priori or calculated as in the previous rule); and (iii) if none of the other rules applies, the population dimension is decreased by a little percentage (1-5%). The growing phases can be implemented either by cloning some selected individuals, by applying genetic operators over them, or by randomly generating novel ones. For population dimension reduction, either the current worst individuals are eliminated or the exclusion proceeds in a random basis, provided that the best element is always preserved.

2.3 Constricted PSO

In a PSO system [8], a group of particles tries to locate the best position in a multidimensional space. Each particle i is described by some parameters, such as its current position vector ($\boldsymbol{x}_i$), which represents a candidate solution, and its fitness value f_i, indicating how good is the solution it represents. At each iteration, the particle swarm moves through the environment, with each particle flying according to its own velocity vector $\boldsymbol{v}_i$. Moreover, each particle i always remembers the best position it has found alone so far, denoted here by $\boldsymbol{p}_i$, and is also aware of the best position found so far by any particle of its topological neighborhood, denoted by $\boldsymbol{p}_g$. Depending on how the neighborhood of each particle is defined, two PSO versions may be derived [8]. In the local version, each particle's neighborhood includes a limited number of particles on its sides, while in the global version of PSO, all particles are perceived as neighbors of each other.

During its flight, each particle performs three basic operations, namely, they evaluate themselves, they compare the quality of their solutions with that of their neighbors, and they try to mimic that neighbor with the best performance so far. This is reflected in the PSO model by accelerating each particle in the direction of its personal best position and also in the direction of the global best position. Thus, the velocity of each particle is updated as [4]:

$$\mathbf{v}_i(t+1)=\chi\big(\mathbf{v}_i(t)+c_1\phi_1\big(\mathbf{p}_i(t)-\mathbf{x}_i(t)\big)+c_2\phi_2\big(\mathbf{p}_g(t)-\mathbf{x}_i(t)\big)\big), \quad (1)$$

where $c_1\phi_1$ and $c_2\phi_2$ allude to stochastic weights delimiting how strongly the i-th particle is attracted by the regions containing $\boldsymbol{p}_i$ and $\boldsymbol{p}_g$, respectively, and χ is a constriction coefficient to restrict the particle to take a smaller updating at each step (usually set as 0.7298). Having the velocity updated, then each particle modifies its position according to the simple rule:

$$\mathbf{x}_i(t+1)=\mathbf{x}_i(t)+\mathbf{v}_i(t+1)\,. \quad (2)$$

3. DYNAMICALLY ADAPTING THE POPULATION SIZE IN PSO

Although varying the population size of any population-based metaheuristics during the run seems to be a rather natural and rewarding approach when implementing this type of algorithm, we feel that the applicability of such strategy outside the EA context has not been deeply investigated so far, particularly regarding the PSO model. The conjecture to be tested here is that the PSO model could profit from the continuous modification of the number of particles available in the long run to perform exploration and exploitation steps, mainly when having to deal with high-dimensional problems.

In order to fill this gap, we have adapted both APGA and PRoFIGA schemes to the PSO context, giving birth to two novel PSO variants named as APPSO and PRoFIPSO, respectively. The main reasons for choosing these algorithms in particular were twofold: (i) they represent two different categories of approaches for tuning control parameters in general [5] (while APGA is classified as an adaptive method, PRoFIGA is deterministic in nature); and (ii) they have outperformed other contestant schemes in the recent comparative study conducted by Eiben et al. [6].

The adaptation of the PRoFIGA model to become compliant with PSO was very straightforward: In the end of each generation, the population dimension is modified according to the same three deterministic rules described in Subsection 2.2. The particles to be removed from the swarm are always the current worst ones, while the new members are generated by cloning good individuals chosen by tournament selection from the current population (the values of the velocity and $\boldsymbol{p}_i$ vectors are reset). As we have implemented the global version of PSO, whenever a particle is created, it is informed about the best position vector found so far by the whole population ($\boldsymbol{p}_g$).

Concerning the APGA scheme, the modification of the basic PSO model had to be more profound as now at each iteration two particles have to be selected to reproduce (via tournament selection) in order to give birth to two novel individuals through a recombination process—remember that there is no recombination operator in the standard PSO model [13]. (We have adopted arithmetic crossover for this purpose as it is tailored to float-point representation [5].) As in PRoFIPSO, whenever a new particle is introduced into the swarm, its $\boldsymbol{v}_i$ and $\boldsymbol{p}_i$ vectors are reset. Furthermore, it is assigned a lifetime value according to the bilinear strategy and then is removed from the pool as soon as its age exceeds its total lifetime. As the fitness value of each particle may vary from one iteration to the other, whenever it improves, the age of the particle is reset to zero and the particle's lifetime value is calculated again according to the new circumstance. Conversely, when the fitness value does not improve, the age of the particle is incremented by one unit.

4. SIMULATIONS

In this section, we report on some of the simulation experiments we have conducted so far in order to evaluate the performance of the two novel PSO variants in long-term runs. In Subsection 4.1, we give details on how the experiments were conducted, that is, which optimization functions were used, which algorithms took place in the contest, and how their control parameters were set. In Subsection 4.2, we present and qualitatively discuss the results.

4.1 Experimental Setting

To assess the potentials of extending the PSO model with the notion of time-varying swarm size, a range of some well-known benchmark test functions [3][9][13][15] have been considered by us. Due to space limitation, we decided to focus here only on those problems that showed to be more difficult to be dealt with by the constricted PSO model, in a manner as to evidence the possible gains incurred by dynamically tuning the population. The four test functions considered are given as follows (referred to as Ackley, Griewangk, Rastrigin, and Rosenbrock, respectively), all with the minimum function value set as zero and with dimension N varying according to the range {10, 20, 50, 100}.

1. $$f_{ack}(\mathbf{x}) = 20 + \exp(1) - 20\exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}\right) - \exp\left(\frac{1}{N}\sum_{i=1}^{N}\cos(2\pi x_i)\right), \; -32.768 \le x_i \le 32.768$$

2. $$f_{gri}(\mathbf{x}) = 1 + \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N}\cos\left(\frac{x_i}{\sqrt{i}}\right), \; -600 \le x_i \le 600$$

3. $$f_{ras}(\mathbf{x}) = \sum_{i=1}^{N}\left(x_i^2 - 10\cos(2\pi x_i) + 10\right), \; -5.12 \le x_i \le 5.12$$

4. $$f_4(x) = \sum_{i=1}^{N}\left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right), \; -2.048 \le x_i \le 2.048$$

In the comparative analysis, besides the standard PSO, three other models were compared against the APPSO and PRoFIPSO variants, namely, the standard GA, APGA, and PRoFIGA, all having individuals encoded with the float-point representation. The standard GA was configured as steady-state, with two-point crossover and simple mutation operators, having 80% and 5% as crossover and mutation rates, respectively. The mating selection was done by employing binary tournaments. The PSO control parameters (χ, c_1, and c_2) were set as mentioned in Subsection 2.3.

As our analysis here concerns primarily the long-term behavior exhibited by the novel algorithms, the stop criterion used for the PSO models was to reach the mark of 80 contiguous generations without improvement in the best fitness value. For the GA models, due to their steady-state character, we let the algorithms run until no improvement was achieved in the best fitness value for 500 generations. Specifically for PRoFIGA and PRoFIPSO, the following parameters were set as indicated [6]: *increaseFactor* of 0.4, *decreaseFactor* of 0.02, V = 500, *minPopulationSize* of 15 and *maxPopulationSize* of 10,000. Specifically for APGA and APPSO, *minLT* = 1 and *maxLT* = 11. All this control parameter value setting was obtained after conducting some preliminary experiments with each of the algorithms and also by taking the values suggested by other authors as reference. There is no claim that they were optimal, however. Finally, for each triad <problem instance, problem dimension, initial population size>, each contestant model was executed fifteen times. The values considered for the initial population size varied according to the range {20, 50, 100}.

4.2 Results and Discussion

Tables 1-4 bring the quantitative results we have obtained considering the different optimization problems and algorithms. In these tables, the figures A, B, C, D, and P refer, respectively, to the mean values of the following variables: the best fitness value found along the search; the generation where the fittest individual appeared for the first time; the final dimension of the population; the number of fitness evaluations necessary to find the fittest individual for the first time; and the initial size of the population.

Table 1 - Results for Ackley function

			Standard			APGA or APPSO				PROFiGA or PROFiPSO			
	P		A	B	D	A	B	C	D	A	B	C	D
GA	20	Average	6,0616	2175	4367	4,7659	2387	8	4792	4,5932	3624	24	6011
		Std. Dev.	1,8457	902	1804	1,0960	697	5	1394	1,4313	880	0	1462
	50	Average	6,3387	2223	4495	5,3111	2286	8	4621	5,1163	3972	39	6805
		Std. Dev.	1,2363	699	1397	1,6286	882	5	1763	1,3391	1219	20	2136
	100	Average	6,4022	2910	5919	4,8983	2230	8	4557	4,7014	3509	49	6179
		Std. Dev.	2,4113	1236	2472	1,3533	810	4	1619	1,8432	1310	26	2400
PSO	20	Average	0,1868	402	8044	0,1540	270	25	14383	0,0000	223	215	954601
		Std. Dev.	0,5015	49	980	0,4065	27	1	1553	0,0000	2	17	18422
	50	Average	0,0000	364	18210	0,2638	260	25	15780	0,0000	225	207	981148
		Std. Dev.	0,0000	11	571	0,5565	29	2	1477	0,0000	3	22	17091
	100	Average	0,0000	335	33533	0,0770	271	25	19738	0,0000	224	203	969796
		Std. Dev.	0,0000	9	928	0,2983	26	1	1433	0,0000	4	23	23950

GA	20	Average	5,3520	3954	7927	4,0422	3983	11	7985	4,2550	6490	24	10998
		Std. Dev.	1,1015	1545	3090	0,9431	1178	5	2357	0,8155	1721	0	2961
	50	Average	5,3699	5774	11595	3,8290	4419	11	8886	4,3729	6561	42	11247
		Std. Dev.	1,3953	1982	3964	0,9332	1341	4	2681	0,8637	1431	17	2609
	100	Average	5,6180	6904	13906	4,0266	4347	12	8792	4,0792	5953	39	10499
		Std. Dev.	1,3907	2308	4616	1,5359	1261	5	2523	0,8276	1116	21	2147
PSO	20	Average	1,5815	551	11028	1,2467	481	27	27266	0,0000	310	37	981712
		Std. Dev.	1,1183	128	2552	0,7391	154	3	9602	0,0000	9	6	16386
	50	Average	0,0770	569	28453	1,1759	503	28	31408	0,0000	311	37	994974
		Std. Dev.	0,2983	59	2930	0,7710	300	7	20386	0,0000	8	8	19850
	100	Average	0,0770	502	50193	1,1983	438	26	31233	0,0000	305	43	983409
		Std. Dev.	0,2983	46	4598	0,8730	106	3	6647	0,0000	9	12	16826

			Standard			APGA or APPSO				PROFiGA or PROFiPSO			
	P		A	B	D	A	B	C	D	A	B	C	D
GA	20	Average	4,8308	9671	19361	3,2784	8297	10	16612	3,2847	14994	31	26617
		Std. Dev.	1,3169	2771	5542	0,6171	1912	3	3824	0,9123	2369	6	4298
	50	Average	5,2870	12390	24828	2,9645	9989	11	20027	3,4779	15968	31	28997
		Std. Dev.	1,5302	4145	8290	0,3342	1586	3	3172	0,5030	3105	8	5991
	100	Average	6,7284	13237	26572	3,1963	9303	9	18704	3,4248	14706	25	27369
		Std. Dev.	1,6638	4157	8313	0,5542	2871	2	5742	0,4177	2356	5	4710
PSO	20	Average	7,9394	1646	32917	0,8783	2592	30	169390	0,1041	997	28	1013810
		Std. Dev.	2,7144	209	4179	0,7868	543	6	37185	0,4031	192	4	12775
	50	Average	3,2011	1002	50077	1,0077	2424	32	163131	0,1433	892	40	1014507
		Std. Dev.	1,1481	92	4619	0,4777	484	7	32758	0,3852	165	7	18112
	100	Average	1,7183	889	88873	1,0346	2484	33	171378	0,0981	896	47	1026823
		Std. Dev.	1,1402	212	21204	0,5778	588	7	40812	0,3798	72	13	15772

GA	20	Average	5,2499	17066	34150	3,2228	15910	7	31838	3,2422	25420	24	47576
		Std. Dev.	0,9092	3667	7333	0,4371	2815	2	5629	0,4811	4088	0	7787
	50	Average	6,1069	22983	46015	3,3837	15058	8	30163	3,5565	26565	24	50496
		Std. Dev.	2,3139	7696	15392	0,4266	3075	1	6149	0,5739	5366	0	10722
	100	Average	11,7753	13597	27292	3,3936	15055	8	30209	3,5086	25837	95	50133
		Std. Dev.	3,1801	6828	13656	0,7157	5630	1	11261	0,4524	4440	15	9459
PSO	20	Average	13,0034	4210	84203	0,6238	5161	50	463062	2,4799	2144	29	1050842
		Std. Dev.	1,4986	484	9684	0,6144	854	8	82012	0,3615	186	3	16625
	50	Average	8,8622	2695	134743	0,5372	5473	57	500967	2,3035	2007	39	1061867
		Std. Dev.	1,1258	361	18043	0,5916	762	17	79072	0,3977	259	7	13674
	100	Average	4,5619	2050	205033	0,3714	5619	49	517220	2,4296	1739	48	1070682
		Std. Dev.	0,9100	153	15319	0,5402	801	16	85481	0,4708	204	17	17406

Table 2 - Results for Griewangk function

			Standard			APGA or APPSO				PROFiGA or PROFiPSO			
	P		A	B	D	A	B	C	D	A	B	C	D
GA	20	Average	2,1489	2489	4996	1,7660	2654	9	5325	2,2764	3674	24	6126
		Std. Dev.	0,9068	964	1928	0,8083	621	6	1242	2,2326	996	0	1676
	50	Average	4,4239	2069	4185	1,7127	2472	11	4993	2,0983	3549	52	6141
		Std. Dev.	2,7578	1153	2307	0,6504	1197	7	2394	1,3806	815	20	1478
	100	Average	2,2774	3129	6357	2,1604	2160	9	4418	1,9876	2998	29	5542
		Std. Dev.	0,8321	1002	2004	1,3918	938	5	1877	0,8519	984	5	1933
PSO	20	Average	0,0917	372	7441	0,0636	250	25	14048	0,0664	262	118	752850
		Std. Dev.	0,0616	79	1579	0,0224	28	1	2590	0,0282	53	86	106586
	50	Average	0,0869	375	18743	0,0700	264	25	15135	0,0593	265	100	813716
		Std. Dev.	0,0440	88	4417	0,0163	44	1	4284	0,0298	38	65	107242
	100	Average	0,0689	349	34887	0,0735	245	25	14658	0,0746	291	91	798937
		Std. Dev.	0,0367	70	7031	0,0207	31	2	2046	0,0245	73	78	101447

GA	20	Average	3,2740	3913	7844	1,5804	4248	9	8515	1,6512	6353	24	10810
		Std. Dev.	1,8174	1304	2608	0,3229	1163	5	2326	0,2776	1549	0	2706
	50	Average	3,2239	5166	10380	2,0716	3751	6	7549	1,9006	6450	51	11264
		Std. Dev.	1,7349	1586	3171	0,7201	1558	0	3116	0,7595	1398	10	2559
	100	Average	3,3706	6617	13331	1,9619	3799	10	7697	1,9539	5752	27	10357
		Std. Dev.	2,1972	1731	3461	0,6382	1206	5	2412	0,6398	850	7	1743
PSO	20	Average	0,0752	558	11159	0,0676	362	27	19399	0,0670	266	101	929544
		Std. Dev.	0,1425	123	2464	0,0504	30	3	1951	0,0406	43	32	30834
	50	Average	0,0103	431	21560	0,0548	371	26	19933	0,0406	257	102	926951
		Std. Dev.	0,0118	35	1772	0,0512	24	3	1425	0,0292	16	34	28434
	100	Average	0,0262	419	41913	0,0595	375	27	21114	0,0640	274	93	940270
		Std. Dev.	0,0196	80	8017	0,0327	21	4	1624	0,0294	44	48	20012

			Standard			APGA or APPSO				PROFiGA or PROFiPSO			
	P		A	B	D	A	B	C	D	A	B	C	D
GA	20	Average	5,0753	8618	17254	1,7978	8915	8	17848	1,7582	13670	28	24330
		Std. Dev.	2,6905	2307	4615	0,3871	1697	2	3394	0,3812	2430	5	4450
	50	Average	8,9815	10174	20396	2,2311	8324	8	16695	2,0439	13824	28	25148
		Std. Dev.	7,2556	3486	6972	0,8131	2570	2	5139	0,5659	1808	5	3510
	100	Average	18,9619	12814	25725	1,9670	8847	9	17792	2,1059	11553	33	21487
		Std. Dev.	26,9309	5175	10349	0,6448	3127	3	6255	0,6486	2116	25	4454
PSO	20	Average	0,4516	2221	44417	0,0061	1227	35	83517	0,0038	629	30	995980
		Std. Dev.	0,5957	222	4449	0,0071	148	5	11272	0,0057	59	3	12956
	50	Average	0,1759	1120	55980	0,0082	1182	33	79805	0,0081	605	38	995757
		Std. Dev.	0,5029	97	4848	0,0108	135	8	10715	0,0114	55	8	13423
	100	Average	0,0434	835	83460	0,0080	1272	33	88792	0,0020	590	53	1006913
		Std. Dev.	0,0715	56	5578	0,0076	201	7	15078	0,0042	78	17	17607

GA	20	Average	8,1313	18220	36458	3,4369	14502	8	29021	2,7163	23929	24	44964
		Std. Dev.	1,6063	2299	4599	1,2815	3819	1	7639	0,5678	3838	0	7477
	50	Average	12,1358	25909	51866	2,8561	16282	7	32613	3,1870	24631	28	47477
		Std. Dev.	10,0629	6409	12819	0,8585	4328	1	8656	1,3655	3428	11	7166
	100	Average	112,6162	15364	30826	3,5640	14719	7	29537	3,1185	22168	83	44095
		Std. Dev.	98,7907	6821	13643	2,5235	3801	2	7603	0,8565	3541	29	8256
PSO	20	Average	2,1373	7219	144373	0,0422	2935	70	327001	0,0277	2804	29	1078338
		Std. Dev.	2,3404	887	17741	0,0405	219	17	28520	0,0361	421	3	18486
	50	Average	0,4278	3692	184583	0,0471	3050	69	343920	0,0378	2557	38	1080199
		Std. Dev.	0,7827	345	17270	0,0559	208	20	30255	0,0537	346	9	18559
	100	Average	0,1076	2330	233020	0,0471	3125	67	353569	0,0211	2364	52	1097606
		Std. Dev.	0,1989	238	23802	0,0492	210	12	35145	0,0273	236	15	13217

In each table, the matrices exhibited at the left side show the results for problem dimensions 10 (top) and 20 (bottom), whereas those displayed at the right side keep the results for problem dimensions 50 (top) and 100 (bottom). In the following, we provide brief comments related to each table in turn, and then conclude the section with some general observations.

By observing Tables 1 and 2, one can notice that P impacts the standard GA and PSO models differently (in general, low values of this variable help the first but not the second), whereas the APPSO and PRoFIPSO models seem to be indifferent to the choice of this variable. As long as the dimension of the problem increases, the performance of the standard PSO model seems to deteriorate, whereas that of the standard GA stagnates. For $N \geq 50$, both APGA and PRoFIGA are more effective than the simple GA, meaning that their values for criterion A have a propensity to be lower. The same fact could be observed regarding the comparison between APPSO and PRoFIPSO against the standard PSO. Indeed, both APPSO and PRoFIPSO were very effective, triumphing over the other methods with respect to the criterion A in the great majority of cases, showing robustness against the choice of N. Unfortunately, the effectiveness of PRoFIPSO was at the expense of high amounts of fitness evaluations to locate the optimal points for the first time. Also, APPSO presents lower values for factor C than PRoFIPSO as long as the size of the problem increases.

According to Table 3, the optimization of the Rastrigin function turns out to be a more difficult problem to be solved by all PSO models. In general, the results of the standard PSO model were

Table 3 - Results for Rastrigin function

	P		Standard A	Standard B	Standard D	APGA or APPSO A	APGA or APPSO B	APGA or APPSO C	APGA or APPSO D	PROFiGA or PROFiPSO A	PROFiGA or PROFiPSO B	PROFiGA or PROFiPSO C	PROFiGA or PROFiPSO D
GA	20	Average	10,7114	2308	4634	9,4970	2252	8	4523	11,3955	3346	24	5604
		Std. Dev.	3,9490	1070	2141	3,9186	929	5	1857	4,1596	1094	0	1861
	50	Average	10,0439	2355	4759	9,9218	1925	12	3897	8,1420	2726	29	4672
		Std. Dev.	2,6801	975	1949	2,2373	861	6	1721	3,2783	984	13	1722
	100	Average	11,0774	2632	5362	9,1588	2053	15	4205	11,9161	3084	36	5657
		Std. Dev.	4,6140	824	1648	4,2608	802	7	1604	6,0321	1084	19	2132
PSO	20	Average	9,2863	416	8329	7,1637	293	24	14415	0,1990	288	78	929886
		Std. Dev.	3,1613	46	924	4,2246	48	1	2355	0,5578	36	37	36337
	50	Average	4,8421	438	21883	6,0361	346	25	17363	0,3317	292	77	935567
		Std. Dev.	2,6018	77	3863	3,3312	94	2	4641	0,6141	36	56	40087
	100	Average	3,0512	435	43460	6,4341	366	27	19551	0,2653	291	74	940889
		Std. Dev.	1,1571	103	10305	3,3608	95	5	4680	0,4554	27	38	32409

	P		Standard A	Standard B	Standard D	APGA or APPSO A	APGA or APPSO B	APGA or APPSO C	APGA or APPSO D	PROFiGA or PROFiPSO A	PROFiGA or PROFiPSO B	PROFiGA or PROFiPSO C	PROFiGA or PROFiPSO D
GA	20	Average	36,2393	9312	18641	21,7956	8741	9	17500	25,3549	13500	32	24031
		Std. Dev.	9,9350	2386	4772	6,1358	2330	3	4661	13,6350	3484	7	6353
	50	Average	41,8591	12245	24538	26,3793	7347	8	14741	27,7179	14087	33	25529
		Std. Dev.	19,5315	3009	6019	8,8385	2119	3	4237	8,8525	2138	9	4152
	100	Average	98,7655	12193	24483	21,8108	8701	8	17500	27,7258	12138	36	22558
		Std. Dev.	156,8512	5257	10515	8,8184	1943	3	3886	7,4244	1792	32	3696
PSO	20	Average	144,4679	1956	39113	7,9124	2548	34	151748	82,5815	817	30	948582
		Std. Dev.	29,7157	253	5055	3,4976	264	8	19435	16,1311	158	4	39951
	50	Average	117,4712	1085	54247	8,5202	2557	31	150121	85,7653	850	38	967012
		Std. Dev.	24,4161	135	6749	4,6900	276	6	18390	12,4395	135	8	39155
	100	Average	107,4553	889	88880	9,9303	2487	32	148312	72,1676	769	50	965700
		Std. Dev.	25,0777	100	9954	3,5625	271	4	18726	16,0426	80	15	49251

	P		Standard A	Standard B	Standard D	APGA or APPSO A	APGA or APPSO B	APGA or APPSO C	APGA or APPSO D	PROFiGA or PROFiPSO A	PROFiGA or PROFiPSO B	PROFiGA or PROFiPSO C	PROFiGA or PROFiPSO D
GA	20	Average	16,9050	4360	8738	11,9095	3913	10	7845	14,3856	6164	24	10456
		Std. Dev.	6,2573	1368	2736	4,5501	993	6	1985	6,2822	1545	0	2681
	50	Average	18,5215	5159	10366	14,8327	4094	11	8236	15,0671	6634	54	11594
		Std. Dev.	6,2809	1752	3503	6,7297	1223	6	2446	3,6640	1755	12	3215
	100	Average	25,0443	5264	10625	12,0873	4417	11	8932	13,3884	5905	29	10649
		Std. Dev.	10,6661	2149	4297	4,2574	1323	5	2645	4,1970	1703	11	3392
PSO	20	Average	35,6858	639	12789	11,4752	713	31	37167	9,6179	372	30	952118
		Std. Dev.	11,7717	113	2268	6,9022	163	7	9272	3,5211	39	3	41213
	50	Average	27,5935	578	28903	12,4038	616	28	32367	9,5516	369	36	961516
		Std. Dev.	9,3673	104	5197	6,4356	107	8	6148	3,9045	44	6	35798
	100	Average	22,8840	523	52347	12,4702	664	34	35993	8,8220	372	48	959301
		Std. Dev.	6,1678	88	8785	6,4136	151	13	8803	3,1880	37	13	41078

	P		Standard A	Standard B	Standard D	APGA or APPSO A	APGA or APPSO B	APGA or APPSO C	APGA or APPSO D	PROFiGA or PROFiPSO A	PROFiGA or PROFiPSO B	PROFiGA or PROFiPSO C	PROFiGA or PROFiPSO D
GA	20	Average	87,8143	17283	34585	46,3791	14362	8	28742	44,3798	23489	25	44054
		Std. Dev.	26,3187	4156	8312	7,1894	2209	1	4418	14,4253	4549	4	8708
	50	Average	102,4129	21205	42458	47,6131	14139	7	28325	48,0532	22399	26	42680
		Std. Dev.	28,5030	4994	9989	16,0648	3636	1	7271	9,6548	4072	7	8163
	100	Average	267,6145	15971	32041	47,3841	14102	7	28301	49,3529	23556	98	46292
		Std. Dev.	154,6776	8026	16052	14,6068	3233	1	6467	11,8733	3243	14	7255
PSO	20	Average	356,1351	6536	130716	10,0194	5861	48	416708	254,7917	2891	29	1009852
		Std. Dev.	45,6314	720	14408	3,5162	564	13	57725	29,3782	408	4	30595
	50	Average	337,7623	3351	167573	8,8330	6254	54	446599	273,4817	2699	36	1022355
		Std. Dev.	50,3034	314	15675	3,7896	554	12	59980	38,3029	326	9	36558
	100	Average	318,7838	2178	217767	8,8928	6121	53	430675	278,4573	2277	47	1059418
		Std. Dev.	44,3966	274	27436	3,0591	407	15	45924	40,2063	277	15	35130

Table 4 - Results for Rosenbrock function

	P		Standard A	Standard B	Standard D	APGA or APPSO A	APGA or APPSO B	APGA or APPSO C	APGA or APPSO D	PROFiGA or PROFiPSO A	PROFiGA or PROFiPSO B	PROFiGA or PROFiPSO C	PROFiGA or PROFiPSO D
GA	20	Average	49,5653	1751	3520	26,5977	2944	11	5907	26,7977	3282	24	5487
		Std. Dev.	22,1544	645	1289	25,9321	2464	6	4929	24,7438	902	0	1521
	50	Average	34,7723	1845	3739	23,1658	2135	8	4317	23,3459	3353	62	5822
		Std. Dev.	30,0833	950	1899	20,9424	753	5	1507	22,9655	1391	6	2424
	100	Average	18,8197	2964	6026	19,1977	3024	13	6145	23,1735	3616	30	6673
		Std. Dev.	10,8998	1130	2261	14,5864	1257	7	2514	20,7822	1285	6	2594
PSO	20	Average	1,0877	6165	123295	0,0000	1837	26	95845	0,0000	2040	29	870688
		Std. Dev.	1,8133	1590	31797	0,0000	144	4	8798	0,0000	191	3	88216
	50	Average	0,5369	5858	292913	0,0000	1865	27	98015	0,0000	1980	41	895485
		Std. Dev.	1,4009	1706	85303	0,0000	191	3	9645	0,0000	263	5	81201
	100	Average	0,0067	5837	583693	0,0000	1802	28	94284	0,0000	1940	49	953571
		Std. Dev.	0,0250	1244	124360	0,0000	177	9	10134	0,0000	122	14	57222

	P		Standard A	Standard B	Standard D	APGA or APPSO A	APGA or APPSO B	APGA or APPSO C	APGA or APPSO D	PROFiGA or PROFiPSO A	PROFiGA or PROFiPSO B	PROFiGA or PROFiPSO C	PROFiGA or PROFiPSO D
GA	20	Average	200,2577	9527	19073	127,9674	12037	9	24093	129,7019	15403	26	27577
		Std. Dev.	73,2469	3802	7603	53,1766	3911	3	7823	51,9453	3059	4	5621
	50	Average	215,2303	10625	21297	128,8894	9773	9	19593	140,8726	15848	33	29143
		Std. Dev.	72,4086	3803	7605	39,8583	2648	3	5296	33,5165	3885	11	7442
	100	Average	366,2880	10228	20554	171,7088	12497	9	25091	152,5693	15461	39	30091
		Std. Dev.	319,6232	5072	10143	60,5492	4818	3	9635	40,9881	4677	28	10021
PSO	20	Average	11,3243	25998	490843	0,0010	15742	36	1132044	0,0001	17717	29	1534381
		Std. Dev.	9,5962	11084	241565	0,0017	1719	9	123420	0,0001	5345	3	165632
	50	Average	4,8085	29057	1452833	0,0004	16406	35	1189166	1,0632	16446	36	1594909
		Std. Dev.	6,1365	9745	487250	0,0007	1997	7	148155	1,8248	4038	8	149154
	100	Average	7,3757	27944	2794433	0,0006	15521	38	1119626	1,9273	14584	49	1693207
		Std. Dev.	19,1399	7292	729201	0,0009	2940	10	219042	3,5445	923	15	47359

	P		Standard A	Standard B	Standard D	APGA or APPSO A	APGA or APPSO B	APGA or APPSO C	APGA or APPSO D	PROFiGA or PROFiPSO A	PROFiGA or PROFiPSO B	PROFiGA or PROFiPSO C	PROFiGA or PROFiPSO D
GA	20	Average	110,5826	3521	7060	63,2096	4718	9	9453	42,9979	6387	25	10872
		Std. Dev.	68,3335	1442	2885	31,2621	1735	5	3471	25,2698	1559	4	2710
	50	Average	75,7410	4343	8733	51,5752	5288	9	10623	41,0131	7061	51	12412
		Std. Dev.	41,6750	1907	3813	29,4258	1898	4	3797	26,8688	2334	12	4259
	100	Average	91,8523	5080	10258	50,5592	5321	10	10741	60,1645	6993	42	12940
		Std. Dev.	104,0731	1960	3919	30,8949	1479	5	2958	33,8214	2028	36	4215
PSO	20	Average	1,4903	11403	228059	0,0000	4383	30	247743	0,2658	4835	30	1071617
		Std. Dev.	4,1625	3321	66426	0,0000	264	7	18984	1,0293	347	4	27302
	50	Average	0,3654	10271	513527	0,2658	4182	26	236530	0,0000	4337	36	1086728
		Std. Dev.	1,0429	1617	80873	1,0293	422	2	25935	0,0000	256	7	39548
	100	Average	0,2840	10751	1075147	0,0000	4247	28	241055	0,0000	3858	52	1094413
		Std. Dev.	1,0252	2001	200120	0,0000	364	5	23841	0,0000	294	18	31666

	P		Standard A	Standard B	Standard D	APGA or APPSO A	APGA or APPSO B	APGA or APPSO C	APGA or APPSO D	PROFiGA or PROFiPSO A	PROFiGA or PROFiPSO B	PROFiGA or PROFiPSO C	PROFiGA or PROFiPSO D
GA	20	Average	450,5086	16568	33153	316,9458	18764	8	37545	17910,1151	7459	2045	14066
		Std. Dev.	164,4343	4321	8641	76,2347	5700	2	11399	11130,9242	12068	2075	22751
	50	Average	538,7215	17089	34226	277,7686	19105	8	38257	10876,3574	15940	1677	31006
		Std. Dev.	130,0150	5278	10557	65,4623	3455	1	6910	11706,1346	13545	2177	26454
	100	Average	24078,6239	431	960	295,6004	18751	8	37600	4812,5722	17552	982	36160
		Std. Dev.	1236,4055	67	135	61,3091	4253	2	8505	9338,3860	12889	1542	26835
PSO	20	Average	77,6638	63429	1268571	0,0296	37868	70	4676134	49,5364	20100	29	1628169
		Std. Dev.	40,4938	39423	788460	0,1108	6655	13	889144	41,5180	19706	4	635414
	50	Average	46,0235	42019	2100940	0,0004	35985	69	4418285	25,8795	27671	35	2035291
		Std. Dev.	26,1471	18948	947387	0,0003	5735	20	715971	34,9492	19220	6	739201
	100	Average	30,8186	51458	5145753	0,0009	39957	71	4945513	15,3290	32689	44	2622476
		Std. Dev.	25,7210	19376	1937613	0,0014	6247	20	811910	28,4723	19335	15	976123

much worse than those achieved by the respective GA models considering high values for N. For low-dimension instances of this problem, the average performance of APPSO could not keep up with that exhibited by PRoFIPSO with respect to factor A, although both display high variance levels. Conversely, for high values of N, the performance of PRoFIPSO was very frustrating and that of APPSO was conversely remarkable. (Note that both methods display similar values for parameter C, which typically does not vary too much across the table.) The contrast between APPSO and APGA also becomes more salient for $N \geq 50$. On the subject of criterion D, PRoFIPSO could not again go on par with APPSO and the other methods. In general, the performance levels exhibited by APGA and PRoFIGA were quite comparable in terms of index A, but APGA needed fewer generations and fitness evaluations on average to pinpoint the best solutions.

Regarding the results displayed in Table 4, it is possible to conclude that, for the Rosenbrock problem, all PSO models could do considerably better than the GA models, concerning factor A alone. For $N \leq 50$, APPSO and PRoFIPSO models exhibited comparable levels of performance, both defeating the standard PSO model. Conversely, for the highest value of N, while APPSO has championed outstandingly, the results provided by PRoFIPSO deteriorate, also displaying high variance. Interesting to notice that, in this circumstance, the index C for APPSO has increased significantly. The same could be said to the figure D, which for APPSO has assumed levels even higher than for PRoFIPSO, when N = 100. This contrasts sharply with what could be observed in general for the other problem instances.

Finally, considering the whole battery of simulation experiments we have conducted so far (even those not presented here), it is possible to conclude that the PSO model can also benefit from the notion of time-varying population sizes, mainly when considering high-dimensional problem instances and for those cases where there are no huge computational costs imposed on the fitness evaluation process. The feeling we have gained is that the dynamic variation of the number of particles, when properly performed, may be influential for bringing about improvements on performance, mainly in terms of the effectiveness issue. In general, the novel PSO variants have outperformed their GA counterparts in most of the problems, whereas APPSO has prevailed, sometimes significantly, over the constricted version of PSO and the PRoFIPSO model for the most difficulty cases (N = 100)—such conclusions are corroborated by statistical tests that could not be presented here.

The definite reasons explaining why the APGA model could not present the same satisfactory levels of performance as APPSO is something that we could not spot yet, although we suspect very much that this has somewhat to do with the particular way APPSO was configured (for instance, with arithmetic crossover instead of two-point crossover). Maybe because APPSO is not plagued by the "lack of dynamism" some recent work claim APGA is predisposed to exhibit [7][10][11]. On the other hand, regarding the comparison between APPSO and PRoFIPSO in terms of the efficiency issue alone, by visually inspecting the anytime behavior [5] exhibited in general by PRoFIPSO, we could usually observe a voluptuous increase in the swarm size taking place in the first generations, followed by a steady decrease of the number of individuals in the course of the search process. This justifies the high values produced by PRoFIPSO for the index D, even though the values of factor C were typically low in magnitude.

5. CONCLUSION

From the set of control parameters available to be calibrated in the PSO model, we have focused in this work on one in particular, the swarm dimension. More specifically, we have investigated the benefits of adapting two well-known population-resizing techniques, APGA and PRoFIGA, originally developed for GAs, to work within the context of the constricted PSO model. The choice for these techniques was deliberate: They represent two different classes of parameter-control approaches, employ completely different mechanisms to control the population size variation, and are flexible enough to be seemingly adapted to other population-based metaheuristics. The two novel PSO variants, namely, APPSO and PRoFIPSO, were evaluated against their GA counterparts and also against the standard GA and PSO models. In general, the APPSO scheme has championed the contest exhibiting good levels of effectiveness and efficiency, particularly when having to cope with high-dimensional problem instances. On the other hand, PRoFIPSO could not accompany well the same levels of efficiency displayed by APPSO, showing sometimes severe deficiencies in terms of scalability.

As future work, we plan to adapt other population-resizing techniques originally developed for EAs [7][10] to run within the PSO context. As well, we shall also examine more deeply how the application of the concept of time-varying population can be advantageous to another population-based metaheuristic, the Differential Evolution (DE), proposed by Storn and Price [15]. Finally, parallel versions of PSO and DE with time-varying homogeneous/heterogeneous demes running concurrently are also in our agenda of investigation [12].

6. ACKNOWLEDGMENTS

The first author gratefully acknowledges the financial support received from CNPq/FUNCAP, through a DCR grant (project number 23661-04).

7. REFERENCES

[1] Arabas, J., Michalewicz, Z. and Mulawka, J. GAVaPS—A genetic algorithm with varying population size. In Proc. of the First IEEE Conference on Evolutionary Computation, pp. 73-78, IEEE Press, 1994.

[2] Bäck, T., Eiben, A.E. and van der Vaart, N.A.L. An empirical study on GAs "without parameters". In Proc. of the 6th Conference on Parallel Problem Solving From Nature, LNCS 1917, pp. 315-324, Springer, 2000.

[3] Carlisle, A. and Dozier, G. An off-the-shelf PSO. In Proc. Workshop Particle Swarm Optimization, 2001.

[4] Clerc, M., Kennedy, J. The particle swarm—explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation, 6(1), 58-73, 2002.

[5] Eiben, A.E. and Smith, J.E. Introduction to Evolutionary Computing, Springer, 2003.

[6] Eiben, A.E., Marchiori, E. and Valkó, V.A. Evolutionary algorithms with on-the-fly population size adjustment. In Proc. of Parallel Problem Solving from Nature, LNCS 3242, pp. 41-50, Springer, 2004.

[7] Fernandes, C. and Rosa, A. Self-regulated population size in evolutionary algorithms, In Proc. of Parallel Problem Solving from Nature, LNCS 4193, pp. 920-929, Springer, 2006.

[8] Kennedy, J., Eberhart, R. C. and Shi, Y. Swarm Intelligence, Morgan Kaufmann, 2001.

[9] Koumousis, V. K., Katsaras, C. P. A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance. IEEE Transactions on Evolutionary Computation, 10(1), 19-28, 2006.

[10] Lobo, F. G. and Lima, C. F. A review of adaptive population sizing schemes in genetic algorithms. In Proc. of the 2005 Workshop on Parameter Setting in Genetic and Evolutionary Algorithms (GECCO), pp. 228-234, 2005.

[11] Lobo, F. G. and Lima, C. F. Revisiting evolutionary algorithms with on-the-fly population size adjustment, In Proc. of Genetic and Evolutionary Computation Conference (GECCO), pp. 1241-1248, ACM Press, 2006.

[12] Minetti, G. and Alfonso, H. Variable size population in parallel evolutionary algorithms, In Proc. of 5th Int. Conference on Intelligent Systems Design and Applications (ISDA), pp. 350-355, IEEE Comp. Soc. Press, 2005.

[13] Shi, Y. and Eberhart, R.C. Empirical study of particle swarm optimization. In Proc. of IEEE Congress on Evolutionary Computing, Vol. III, pp 1945-1950, 1999.

[14] Smith, R.E. and Smuda, E. Adaptively resizing populations: Algorithm, analysis, and first results. Complex Systems, 9, pp. 47–72, 1995.

[15] Price, K., Storn, R. and Lampinen, J. Differential Evolution: A Practical Approach to Global Optimization, Springer, 2005.