Exploration in stochastic algorithms: An application on $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System

Paola Pellegrini, Daniela Favaretto and Elena Moretti

Abstract In this paper a definition of the exploration performed by stochastic algorithms is proposed. It is based on the observation through cluster analysis of the solutions generated during a run. The probabilities associated by an algorithm to solution components are considered. Moreover, a consequent method for quantifying the exploration is provided. Such a measurement is applied to $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System. The results of the experimental analysis allow to observe the impact of the parameters of the algorithm on the exploration.

Keywords: exploration, cluster analysis, $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System.

1 Introduction

In the last decades, stochastic algorithms have been extensively used for tackling combinatorial optimization problems. Owing to their success, they have been object of several studies, both computational and theoretical. As for the former are concerned, the papers present many applications of algorithms to classical optimization problems and to their variants [1, 2, 3]. As for the latter, the studies focus mainly on the quality of the solution provided and on the convergence behavior of the approaches [4, 5, 6]. This study moves away from both these branches of the literature: Its aim is dealing with the exploration of the search space.

The current study proposes a definition and an approach for measuring the exploration performed by a stochastic algorithm during a run. Although the interest on this topic is quite intuitive, few papers focus on these elements [7, 8, 9, 10, 11]. In none of them this matter is treated from a general point of view. Instead, the main characteristic that the definition of exploration must have is being independent on the specific procedure considered: given a problem, an instance and some infor-

Paola Pellegrini, Daniela Favaretto, Elena Moretti
Department of Applied Mathematics, Dorsoduro 3825/E, I-30123 Venezia, Italy
e-mail: paolap@pellegrini.it, favaret@unive.it, emoretti@unive.it

mation on a run, one must be able to measure the exploration performed. Such a measure needs to be coherent with the behavior of the algorithm: In other words, one cannot just focus on the topological space, locating there solutions and deciding a rule according to which to state whether the procedure has explored deeply. For doing that, after fixing a representation of solutions, one should define a distance measure, independently on the procedure to be evaluated.

After giving such a definition of exploration, the consequent quantification method is applied to $\mathscr{MAX}$–$\mathscr{MIN}$ Ant System [12, 13, 14, 15], an ant colony optimization (ACO) algorithm. In particular, its main parameters are considered, and their impact on the exploration performed is analyzed. Instances of the traveling salesman problem are considered.

The rest of the paper is organized as follows. In Section 2 the importance of the understanding of the explorative behavior of stochastic algorithms is depicted. Section 3 proposes a representation of the space of solutions, a definition of exploration, and a consequent measurement method. Section 4 shortly describes the $\mathscr{MAX}$–$\mathscr{MIN}$ Ant System algorithm and the main elements needed for the application of such method. In Section 5 the experimental analysis is reported, and the results are discussed. Finally, Section 6 concludes the paper.

2 The relevance of understanding exploration in stochastic algorithms

While the theoretical interest in understanding the exploration performed by an algorithm is quite intuitive, the practical relevance may be more muddled. On one hand, it may help in understanding why some procedures work better than others on some specific problems. On the other hand, some knowledge on the exploration may make a little bit more scientific the choice of some algorithmic components.

Suppose one needs to deal with a problem whose formulation and feasible region are known. In general, the solution proposed to the problem "selection and implementation of an algorithm for solving some instances" will not be a global optimum. The feasible region is too wide for being exhaustively investigated. Some region that appears more familiar will be strongly attractive. Most researchers and practitioners are somehow specialists of one or few procedures. This leads them to choose it – or one of the few – for tackling any problem, unless some other approach appears significantly better in the single case. In general, such a clear ranking is not evident *a priori*. So, suppose this problem of algorithm selection is avoided.

Even at this point, the problem is not yet solved: Whatever the approach chosen, in the literature several modules for improving its performances are proposed [15, 16, 17, 18, 19, 20, 21, 22, 23, 24]. Some of them can be used together, some others are alternative. The approach used in general is implementing a few variants of the algorithm, running some "preliminary experiments" and choosing the version that appears to perform best. The sub-optimality of this procedure is quite clear. Nonetheless, the experience on the implication of each module is the only help in

trying to get to, at least, a local optimum. The same reasoning, on a smaller scale, may be done for the choice of the values to assign to the parameters of the procedure. Some tuning procedures are available in the literature [25, 26, 27], but in general the results they produce are affected by some starting point decided by the implementer. The problem is again how to choose this starting point.

The intuition of the implementer will never be replaced by some computation. Nonetheless, it may be required at a higher level: It may be restricted to the level of exploration needed. An indication on the type of exploration allowed by specific modules would make easier the choice between them. For clarifying the idea at the basic level, suppose the algorithm has already been implemented. Only the parameters need to be set. Suppose the value of one of them has to be between 0 and 1. Moreover, a tuning procedure is available. It allows to choose "the best value" in a predefined range. Suppose resources allow to test only 10 values. Should the whole set $[0,1]$ be considered, with consequent steps of 0.1? Or should the set $[0,0.5]$ be analyzed with steps of 0.05? It depends on what is known on the impact of the value of this parameter on the exploration performed, and on what is the level of exploration desired. This knowledge allows a finer search in the suitable interval, or at least it makes the process faster by allowing the *a priori* elimination of some value.

3 Exploration: a definition

Let a combinatorial optimization problem be mapped on a graph $G = (N,A)$, where N is the set of nodes and A the set of edges, and $|N| = n$ and $|A| = a$. A solution is a vector of a components. In particular, solution S is given by $S = \{x_1, x_2, ..., x_a\}$. $x_i = 0$ if the i-th edge is not included in it, and $x_i = P_i$ otherwise: Let P_i be the probability the algorithm assigns to that edge when constructing that solution. Since stochastic algorithms are considered, this probability is always computable, in a different way for each different approach. In this sense, solutions are observed from the point of view of the algorithm. Such a representation of solutions has been chosen for reflecting as much as possible the characteristics of stochastic algorithms.

In order to observe the distribution of solutions in the space $\Re^a$, it is necessary to define a distance between solutions. The main requirements for this measure are the following: The distance between two solutions that share all the edges must be null regardless their probabilities. The distance between two solutions that differ for some edges must depend on their probabilities. Let S_1 and S_2 be two solutions ($S_1 = \{x_1, x_2, ..., x_a\}$, $S_2 = \{y_1, y_2, ..., y_a\}$). Let $d_i = x_i - y_i$, if $x_i y_i = 0$, and $d_i = 0$ otherwise. The distance between S_1 and S_2 is

$$D(S_1, S_2) = \sqrt{\sum_{i \in A} (d_i)^2}. \quad (1)$$

The distance between two solutions, then, lays in the interval $\left[0, \sqrt{2z}\right)$, where z is the number of edges included in a solution. The upper bound is defined considering two solutions made by z different edges with probabilities that tend to 1.

According to this representation, it is possible to group solutions in clusters [28, 29, 30]. An agglomerative hierarchical procedure [31] is considered: At each step, the two closest solutions are grouped together to form a cluster. The distance between this new cluster and the others, is the maximum distance [32] between the two just unified clusters and the others. The aggregative procedure stops when the distance between the two closest clusters is greater than a predefined threshold ε. For the purpose of this study, solutions that are grouped in a cluster can be treated as a unique solution.

In this framework, it is possible to define the exploration:

Definition 1. The exploration performed by a stochastic algorithm in a run is given by the number of clusters built.

In order to compute this value one only needs to know how the stochastic algorithm associates probabilities to edges. As an application of this concept, from here on, an ACO algorithm is studied.

4 The case of ant colony optimization

$\mathscr{MAX}$–$\mathscr{MIN}$ Ant System is one of the best performing ant colony optimization algorithms [12, 13, 14, 15]. In $\mathscr{MAX}$–$\mathscr{MIN}$ Ant System the pheromone update is applied after the activity of each colony of ants modifying the pheromone on each edge $i \in A$, τ_i, according to

$$\tau_i = (1-\rho)\tau_i + \Delta\tau_i^b, \tag{2}$$

where $\Delta\tau_i^b = 1/C_b$ if edge i belongs to the best solution b, and $\Delta\tau_i^b = 0$ otherwise. C_b is the cost associated with solution b. Solution b is either the iteration-best solution or the best-so-far solution. The schedule according to which the solution to be used is chosen, is described by [15].

The pheromone trail in $\mathscr{MAX}$–$\mathscr{MIN}$ Ant System is bounded between τ_{MAX} and τ_{min}. Following [15], $\tau_{MAX} = 1/(\rho C_{best-so-far})$, and $\tau_{min} = [\tau_{MAX}(1-\sqrt[n]{0.05})]/[(\frac{n}{2}-1)\sqrt[n]{0.05}]$. At the beginning of a run, the best solution corresponds to the one found by a problem specific heuristic. The pheromone is set equal to τ_{MAX} on all the edges.

Another element characterizing ACO algorithms is the random-proportional rule. In particular, let ant k be in node $u \in N$, and let $N_k \subset N$ be the set of nodes not visited yet. Moreover let $A_k^u \subset A$ be the set of edges going from u to nodes in N_k. Each edge $i \in A_k^u$ has a probability of being chosen p_i described in the random proportional rule:

$$p_i = \frac{[\tau_i]^\alpha [\eta_i]^\beta}{\sum_{j \in A_k^u} [\tau_j]^\alpha [\eta_j]^\beta}, \tag{3}$$

where η_i is a heuristic measure associated with edge i [15].

Such a definition of probabilities may neglect the information on the attractiveness of the edges. Consider, for example, the case of the last edge chosen: its probability is 1, independently on its desirability. In the computation of exploration, this situation is an inconvenience. In fact, the values of probabilities may depend more on randomness that affects solutions construction than on the real indication of the algorithm.

In order to better represent this indication, to each edge $i \in A$ a probability P_i is associated:

$$P_i = \frac{[\tau_i]^\alpha [\eta_i]^\beta}{\sum_{j \in A} [\tau_j]^\alpha [\eta_j]^\beta}. \tag{4}$$

In the following an experimental analysis on the traveling salesman problem is proposed.

5 Experimental analysis

In this section the impact of the values of the parameters on the exploration performed by $\mathscr{MAX}$–$\mathscr{MIN}$ Ant System is analyzed. The ACOTSP program implemented by Thomas Stützle is considered [15]. The code has been released in the public domain and is available for free download on www.aco-metaheuristic.org/aco-code/. The stopping criterion considered is the fulfillment of 1000 objective function evaluations. In a deeper experimental analysis this horizon will be widened. At this early stage, no local search procedure is applied. The code used for computing the exploration is available on the web page http://www.paola.pellegrini.it.

Three TSP instances are used. They are generated through `portgen`, the instance generator adopted in the DIMACS TSP Challenge. In particular, they consist of two dimensional integer-coordinate cities grouped in clusters that are uniformly distributed in a square of size $10^6 \times 10^6$. Each instance includes 50 nodes. They are available on the web page www.paola.pellegrini.it.

The parameters analyzed are α, β, ρ, m. The values suggested in the literature [15] $\alpha = 1, \beta = 3, \rho = 0.02, m = 50$ are considered. Then, one parameter at a time is varied. The values used for parameters are reported in Table (1). The number of values considered for α and β is smaller than in the other two cases: Four elements are sufficient for observing evident trends. For each set, the three instances are solved. The solutions built are recorded, together with the probabilities associated to edges at each iteration. Then, the distance matrix of the solutions are computed according to 1. The number of clusters in which they are grouped is evaluated. In the three cases, the value of ε is varied between the lower and the upper bound of distance among solutions, with steps of 0.5. In particular, values between 0.5 and 10

Table 1 Values tested for parameters

parameter	values
α	1, 2, 3, 4
β	3, 4, 5, 6
ρ	0.02, 0.05, 0.1, 0.2, 0.25, 0.3, 0.5, 0.7, 0.9
m	10, 20, 30, 50, 70, 100

Table 2 Exploration for different values of α and ε.

	First instance					Second instance					Third instance				
ε	α				ε	α				ε	α				
	1	2	3	4		1	2	3	4		1	2	3	4
0.5	1000	1000	1000	1000	0.5	1000	1000	1000	1000	0.5	1000	1000	1000	1000
1.0	995	1000	1000	999	1.0	983	988	985	990	1.0	995	997	997	997
1.5	820	857	872	899	1.5	688	731	754	796	1.5	649	718	738	816
2.0	378	409	441	488	2.0	293	329	350	398	2.0	215	256	298	369
2.5	140	157	184	228	2.5	111	134	153	187	2.5	51	73	90	132
3.0	45	55	65	89	3.0	40	49	63	75	3.0	10	20	29	44
3.5	10	13	23	31	3.5	13	15	21	30	3.5	3	5	7	13
4.0	2	2	6	9	4.0	4	6	7	11	4.0	1	2	2	3
4.5	1	1	1	2	4.5	1	1	1	2	4.5	1	1	1	1
5.0	1	1	1	1	5.0	1	1	1	1	5.0	1	1	1	1
5.5	1	1	1	1	5.5	1	1	1	1	5.5	1	1	1	1
6.0	1	1	1	1	6.0	1	1	1	1	6.0	1	1	1	1

are used. The distance between solutions varies in the interval $[0, \sqrt{2n} = 10)$ since a solution includes n edges.

The results are reported in Figures 1 and 2 and in Tables 2, 3, 4, 5. The only measure considered is the level of exploration performed. The quality of the solutions generated is neglected, being out of the scope of this analysis. They are grouped according to the parameter being changed. In the graphics, the x-axis represents the variation of ε, while the y-axis the number of clusters, i.e. the exploration. The maximum value reported for ε is 6. In fact, for $\varepsilon \geq 6$ the exploration is equal to 1 no matter what the value of the parameters are. The range of the exploration goes from 1 to 1000, the number of solutions generated. Clearly, the results point out that the exploration is a decreasing function of ε. Moreover, it is to be remarked that the conclusions that can be drawn on the impact of the parameters on the exploration are the same whatever value we consider for ε, provided that it does not imply extreme values (exploration equal either to 1000 or to 1).

Figure 1 and Tables 2, 3 concern parameters α and β. In both cases the exploration is an increasing function of the value of the parameter. The reason of this trend can be found by observing the relation between the variations of the pheromone values and the consequent modifications of the probabilities associated to edges: Let us consider the ratio between the probabilities associated to two edges i and j at a specific time:

$$\frac{p_i}{p_j} = \frac{P_i}{P_j} = \left[\frac{\tau_i}{\tau_j}\right]^{\alpha} \left[\frac{\eta_i}{\eta_j}\right]^{\beta}. \tag{5}$$

Table 3 Exploration for different values of β and ε.

First instance

ε	β			
	3	4	5	6
0.5	1000	1000	1000	999
1.0	995	997	998	994
1.5	820	924	945	947
2.0	378	584	656	704
2.5	140	282	355	398
3.0	45	124	172	210
3.5	10	50	77	105
4.0	2	14	30	43
4.5	1	3	10	13
5.0	1	1	2	3
5.5	1	1	1	1
6.0	1	1	1	1

Second instance

ε	β			
	3	4	5	6
0.5	1000	1000	1000	1000
1.0	983	995	997	995
1.5	688	853	921	929
2.0	293	478	601	658
2.5	111	233	320	382
3.0	40	111	170	217
3.5	13	47	79	113
4.0	4	18	33	49
4.5	1	5	13	17
5.0	1	1	2	5
5.5	1	1	1	1
6.0	1	1	1	1

Third instance

ε	β			
	3	4	5	6
0.5	1000	1000	1000	1000
1.0	995	999	998	999
1.5	649	896	930	940
2.0	215	462	576	622
2.5	51	183	270	332
3.0	10	61	124	151
3.5	3	19	45	64
4.0	1	6	17	26
4.5	1	1	4	8
5.0	1	1	1	1
5.5	1	1	1	1
6.0	1	1	1	1

Table 4 Exploration for different values of ρ and ε.

First instance

ε	ρ								
	0.02	0.05	0.1	0.2	0.25	0.3	0.5	0.7	0.9
0.5	1000	1000	1000	1000	1000	955	899	825	680
1.0	995	996	1000	997	997	922	825	755	590
1.5	820	853	916	965	957	852	725	653	481
2.0	378	422	539	702	700	657	552	476	348
2.5	140	171	241	371	401	385	329	316	219
3.0	45	58	106	184	216	209	182	186	122
3.5	10	16	35	86	98	101	91	109	66
4.0	2	3	11	34	41	44	47	57	35
4.5	1	1	3	12	17	18	22	25	17
5.0	1	1	1	5	6	7	9	12	8
5.5	1	1	1	1	3	2	4	5	4
6.0	1	1	1	1	1	1	2	2	2

Second instance

ε	ρ								
	0.02	0.05	0.1	0.2	0.25	0.3	0.5	0.7	0.9
0.5	1000	1000	1000	1000	999	951	843	826	850
1.0	983	990	990	996	984	901	759	717	712
1.5	688	739	851	935	920	819	627	576	542
2.0	293	354	463	659	654	611	455	413	380
2.5	111	144	210	367	373	355	284	261	232
3.0	40	57	99	206	202	197	155	153	135
3.5	13	21	39	100	106	99	87	81	69
4.0	4	7	15	46	48	52	42	45	35
4.5	1	1	4	19	22	25	19	18	20
5.0	1	1	1	7	6	9	10	10	9
5.5	1	1	1	1	2	4	4	5	4
6.0	1	1	1	1	1	1	2	2	1

Third instance

ε	ρ								
	0.02	0.05	0.1	0.2	0.25	0.3	0.5	0.7	0.9
0.5	1000	1000	1000	1000	999	983	797	804	667
1.0	995	998	1000	998	993	947	739	705	567
1.5	649	749	825	947	918	837	643	579	456
2.0	215	289	409	627	657	545	480	417	311
2.5	51	81	159	337	374	302	286	248	184
3.0	10	21	56	156	177	147	162	135	109
3.5	3	6	19	64	84	73	87	68	61
4.0	1	1	6	24	37	27	43	35	30
4.5	1	1	1	9	14	13	17	18	16
5.0	1	1	1	3	5	5	8	8	8
5.5	1	1	1	1	2	2	4	4	4
6.0	1	1	1	1	1	1	1	2	2

Suppose, with no loss of generality, that $\eta_i > \eta_j$. The further the value of (5) from 1, the stronger the attractiveness of edge i with respect to j. During a run of the algorithm, the only variable element in (5) is the ratio between pheromone values. A high value of α implies that a slight change in this ratio, going from slightly below to slightly above 1 (or viceversa), implies a strong variation of the ratio between the probabilities. A similar reasoning can be made for β: a high value of this parameter amplifies a lot any variation of τ_i/τ_j. As a consequence, the lower these parameters,

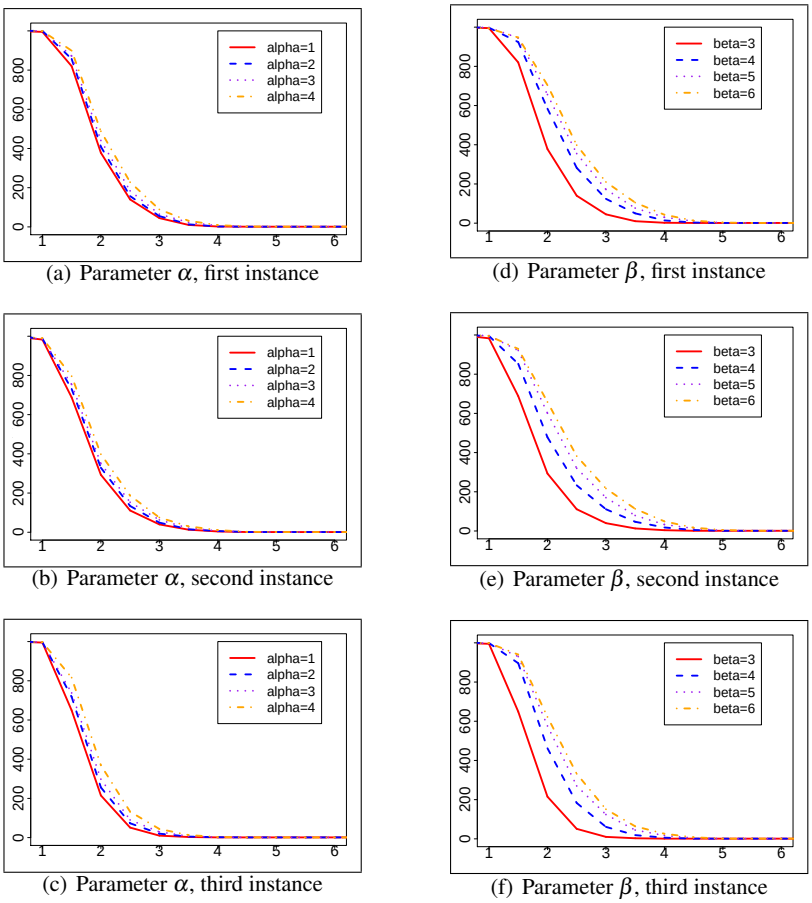
Table 5 Exploration for different values of m and ε.

First instance							Second instance							Third instance						
ε	m						ε	m						ε	m					
	10	20	30	50	70	100		10	20	30	50	70	100		10	20	30	50	70	100
0.5	1000	1000	999	1000	999	1000	0.5	1000	1000	999	1000	999	1000	0.5	1000	1000	999	1000	999	1000
1.0	998	996	997	995	995	996	1.0	993	993	991	983	988	982	1.0	999	994	996	995	997	993
1.5	924	867	828	820	789	798	1.5	839	745	698	688	664	670	1.5	838	751	680	649	626	610
2.0	550	439	398	378	360	350	2.0	454	359	308	293	295	275	2.0	429	295	244	215	210	183
2.5	260	182	160	140	133	124	2.5	215	147	121	111	122	108	2.5	159	96	70	51	61	44
3.0	107	62	50	45	32	35	3.0	97	57	43	40	36	36	3.0	56	27	14	10	12	9
3.5	41	19	14	10	7	7	3.5	42	20	12	13	10	10	3.5	19	8	5	3	4	2
4.0	13	4	3	2	1	1	4.0	16	6	4	4	3	2	4.0	6	3	1	1	1	1
4.5	3	1	1	1	1	1	4.5	4	1	1	1	1	1	4.5	1	1	1	1	1	1
5.0	1	1	1	1	1	1	5.0	1	1	1	1	1	1	5.0	1	1	1	1	1	1
5.5	1	1	1	1	1	1	5.5	1	1	1	1	1	1	5.5	1	1	1	1	1	1
6.0	1	1	1	1	1	1	6.0	1	1	1	1	1	1	6.0	1	1	1	1	1	1

the less extreme the short term variation of the ratio between probabilities. Given that they need to sum to one, this implies that, the lower these parameters, the less extreme the probabilities, i.e. the further from 0 and 1. From this, it comes that the distance between solutions is in general smaller, and, then, the exploration is lower. For visualizing this observation, one may think that with a low value of α (or β), ants need the ratio between τ's to be very high before being forced to move to a specific region of the search space. Once such a high ratio is reached, it may be very hard, and it may take a long time, to update the pheromone in such a way that the edges that are very attractive now can be neglected. In this long period, ants will not go far from already visited areas. If α is high, instead, even one single deposit of pheromone may imply a great modification of the ratios between probabilities.

Figures 2(a), 2(b) and 2(c) and Table 4 report the results achieved when varying ρ. The relation between this parameter and the exploration is more complicated than in the previous case. Up to a certain threshold (that in these cases is around 0.25), the exploration is an increasing function of ρ. Beyond this threshold, the exploration is a decreasing function of this parameter. The first trend may be due to the fact that, for very low values of ρ, the amount of pheromone deposited and its evaporation are very low as well. As a consequence, several updates may be necessary before changing the ranking of the attractiveness of edges: Several updates will be necessary before changing the area to be investigated with high probability. In this low range, then, the higher ρ, the higher the exploration. On the other hand the situation changes once the threshold is passed: An increase of the value of this parameter implies that, even with one single update, the evaporation on the edges that are not used is very high. The same applies to the deposit on the ones included in the best solution. The latter edges, then, become much more attractive than the others in a very short time. Solutions including different edges are very seldom constructed. The higher the value of ρ the stronger this effect, and then the smaller the distance between solutions and the consequent exploration.

Finally, Figures 2(d), 2(e) and 2(f) and Table 5 concern parameter m. In this case, the exploration appears decreasing in the value of the parameter. This may be

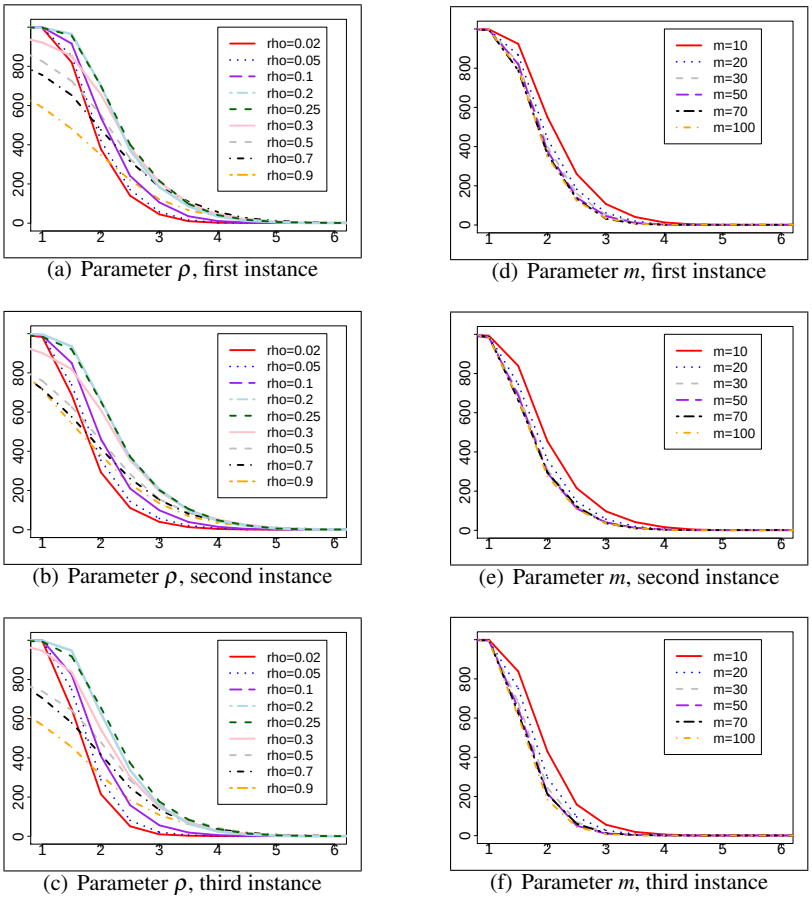


(a) Parameter α, first instance
(d) Parameter β, first instance
(b) Parameter α, second instance
(e) Parameter β, second instance
(c) Parameter α, third instance
(f) Parameter β, third instance

Fig. 1 Number of clusters as a function of ε when varying parameters α and β.

explained by observing that the greater m, the lower the number of iterations performed in a run, and then the lower the number of different probability distributions used. A solution composed by edges with low probabilities, is very unlikely to be constructed. Moreover, if only few probability distributions are used, it is quite unlikely that such edges will ever have a high probability, and then will ever be part of a solution constructed. In this sense, a high value of m implies that the edges used are often the same, and then that the distance between solutions is small. It follows that the higher m, the lower the exploration.

In [33] exploration was introduced only from an intuitive point of view. It was intended as the distribution of solutions in promising areas of the search space. This interpretation may be seen as the distribution of solutions inside clusters. The analysis of this fact may be object of future research, but nothing appears to be really in contrast with this experimental analysis.



(a) Parameter ρ, first instance (d) Parameter m, first instance

(b) Parameter ρ, second instance (e) Parameter m, second instance

(c) Parameter ρ, third instance (f) Parameter m, third instance

Fig. 2 Number of clusters as a function of ε when varying parameters ρ and m.

6 Conclusions

In this paper a definition of the exploration of the search space performed by a stochastic algorithm is proposed. Its main distinction with respect to what is present in the literature, is that it can be quite easily applied to any stochastic algorithm. It is based on cluster analysis: the distribution of solutions found is evaluated. The space in which they are studied is the space of the probabilities that the algorithm itself associates to solution components. The consequent quantification measure presented allows to make fair comparison on the exploration performed by different approaches.

Moreover, an application of the measurement method is reported. It is based on $\mathscr{MAX}$–$\mathscr{MIN}$ Ant System and on the traveling salesman problem. The computations reported allow to observe some relations between the the values of the

parameters of the algorithm and the exploration performed. Such an understanding will allow to better analyze the behavior of the procedure. More practically, it will allow to properly set the parameters, once the exploration needed in a run is decided. In these first experiments, only one parameter at a time is varied, and no local search is applied. Both these points need to be object of further analysis. In particular, it is expectable that the interaction between parameters has an impact on the exploration performed. Such an interaction may come up, for example, in one parameter taking the upper hand against the others, or in two parameters having similar weights, and then pushing in the same (or opposite) direction on the level of exploration. These limitations will be overcome in future research.

Another element that will be object of investigation is the relation between the measure of exploration proposed and other - algorithm specific - methods presented in the literature. Among the others, for ant colony optimization, the average branching factor [15] may be considered for a comparison.

Moreover, a very interesting point to clarify is the link between exploration of the algorithm and quality of the solutions found.

References

1. L. Bianchi, M. Birattari, M. Chiarandini, M. Manfrin, M. Mastrolilli, L. Paquete, O. Rossi-Doria, T. Schiavinotto, Journal of Mathematical Modelling and Algorithms **9**(1), 91 (2006)
2. L.A.N. Lorena, J.C. Furtado, Evolutionary Computation **9**(3), 309 (2001)
3. A. Rizzoli, R. Montemanni, E. Lucibello, L. Gambardella, Swarm Intelligence **1**, 135 (2007)
4. W.J. Gutjahr, G. Sebastiani, Methodology and Computing in Applied Probability (2008). To appear
5. S. Hanafi, Journal of Heuristics **7**(1), 47 (2001)
6. T. Stützle, H.H. Hoos, in *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, ed. by S. Voss, S. Martello, I.H. Osman, C. Roucairol (Kluwer Academic Publisher, Dordrecht, The Netherlands, 1999), pp. 137–154
7. H. Esbensen, E. Kuh, in *EURO-DAC '96/EURO-VHDL '96: Proceedings of the conference on European design automation* (IEEE Computer Society Press, Los Alamitos, CA, USA, 1996), pp. 356–361
8. M.L. Maher, J. Poon, Microcomputers in Civil Engineering pp. 195–210 (1996)
9. P. Pellegrini, Aco: parameters, exploration and quality of solutions. Ph.D. thesis, Università Ca' Foscari, Department of Applied Mathematics, Venice, Italy (2007)
10. M. Toussaint, in *2003 Genetic and Evolutionary Computation Conference (GECCO 2003)* (2003), pp. 1444–1456
11. J.P. Watson, L.D. Whitley, A.E. Howe, Journal of Artificial Intelligence Research **24**, 221 (2005)
12. T. Stützle, H.H. Hoos, in *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC'97)*, ed. by T. Bäck, Z. Michalewicz, X. Yao (IEEE Press, Piscataway, NJ, USA, 1997), pp. 309–314
13. T. Stützle, H.H. Hoos, in *Artificial Neural Networks and Genetic Algorithms*, ed. by R.F.A. G. D. Smith, N. C. Steele (Springer Verlag, Vienna, Austria, 1998), pp. 245–249
14. T. Stützle, H.H. Hoos, Future Generation Computer Systems **16**(8), 889 (2000)
15. M. Dorigo, T. Stützle, *Ant Colony Optimization* (MIT Press, Cambridge, MA, USA, 2004)
16. D. Corne, M. Dorigo, F. Glover (eds.), *The Ant Colony Optimization Meta-Heuristic* (McGraw-Hill, New York, NY, USA, 1999)

17. F. Glover, G. Kochenberger (eds.), *Handbook of Metaheuristics* (Kluwer Academic Publisher, Norwell, MA, USA, 2002)
18. D.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning* (Addison-Wesley Publishing Company, Reading, MA, USA, 1989)
19. H.H. Hoos, T. Stützle, *Stochastic Local Search. Foundations and Applications* (Morgan Kaufmann Publishers, San Francisco, CA, USA, 2004)
20. T. Ibaraki, K. Nonobe, M. Yagiura (eds.), *Metaheuristics: Progress as Real Problem Solvers* (Kluwer Academic Publisher, Boston, MA, USA, 2005). Post-conference proceedings of the 5th Metaheuristics International Conference, MIC 2003
21. P. Larrañaga, J.A. Lozano, *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation* (Kluwer Academic Publisher, Boston, MA, USA, 2001)
22. Z. Michalewicz, D.B. Fogel, *How to Solve it: Modern Heuristics* (Springer Verlag, Berlin, Germany, 2000)
23. I.H. Osman, J.P. Kelly (eds.), *Meta-Heuristics: The Theory and Applications* (Kluwer Academic Publisher, Boston, MA, USA, 1996). Post-conference proceedings of the 1st Metaheuristics International Conference, MIC'95
24. S. Voss, S. Martello, I.H. Osman, C. Roucairol (eds.), *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization* (Kluwer Academic Publisher, Boston, MA, USA, 1999). Post-conference proceedings of the 2nd Metaheuristics International Conference, MIC'97
25. R. Battiti, in *Modern Heuristic Search Methods*, ed. by V.J. Rayward-Smith, I.H. Osman, C.R. Reeves, G.D. Smith (John Wiley & Sons, Chichester, United Kingdom, 1996), pp. 61–83
26. M. Birattari, T. Stützle, L. Paquete, K. Varrentrapp, in *Proceedings of the Genetic and Evolutionary Computation Conference*, ed. by W.B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M.A. Potter, A.C. Schultz, J.F. Miller, E. Burke, N. Jonoska (Morgan Kaufmann Publishers, San Francisco, CA, USA, 2002), pp. 11–18
27. J. Xu, S.Y. Chiu, F. Glover, International Transactions in Operational Research **5**(3), 233 (1998)
28. B. Everitt, *Cluster Analysis* (Heinemann Educational Books, London, UK, 1974)
29. S.L. B. Everitt, M. Leese, *Cluster Analysis* (Arnold, London, UK, 2001)
30. L. Kaufman, P.J. Rousseeuw, *Finding groups in data. an introduction to cluster analysis* (Wiley Series in Probability and Mathematical Statistics. Applied Probability and Statistics, New York: Wiley, 1990, 1990)
31. N. Jardine, R. Sibson, The Computer Journal **11**(2), 177 (1968)
32. M.J.A. Berry, L. G., *Mastering Data Mining* (Wiley Computer Publishing, 2000)
33. P. Pellegrini, D. Favaretto, E. Moretti, in *Ant Colony Optimization and Swarm Intelligence, 5th International Workshop, ANTS 2006*, *LNCS*, vol. 4150, ed. by M. Dorigo, L. Gambardella, M. Birattari, A. Martinoli, R. Poli, T. Stützle (Springer Verlag, Berlin, Germany, 2006), *LNCS*, vol. 4150, pp. 203–214