Online PSO for Web Marketing Optimization

Alfredo Milani, Valentino Santucci
Department of Mathematics and Computer Science
University of Perugia
via Vanvitelli, 1, Perugia, Italy
milani@unipg.it, valentino.santucci@dmi.unipg.it

Abstract— **In this paper we propose an approach to optimization of web marketing content based on an online particle swarm optimization (PSO) model. The idea behind online PSO is to evaluate the collective user feedback as the PSO objective function which drives particles velocities in the hybrid continuous-discrete space of web content features. PSO coordinates the process of sampling collective user behavior in order to optimize the web marketing metric. To improve the performances a variation to the PSO schema is adopted, this variation consists in a restart of the algorithm if the convergence speed is not good. Experiments in the scenario of the home page of an online shop show that the method converges faster and avoid some common drawbacks such as local optimal and hybrid discrete/continuous features management; however is observed that the restart procedure improves the convergence speed of some difficult instances of the problem without affects the other ones. The proposed online optimization method is general and can be applied to other web marketing or business intelligent contexts.**

Keywords: web marketing optimization, collective behavior mining, collaborative intelligence, online particle swarm optimization

I. INTRODUCTION

The optimization of web content presentation [11, 12] is today a very hot issue for e-commerce applications. Whether considering web pages, advertising banners or any other content presentation media on the web, choosing the right layout and appearance combination, with respect to the given target of users, can obtain more effective and successful impact, such as gathering more readers to web sites, customers to online shops or clicks to advertising banners.

Let consider the optimization of the home page of an e-commerce web site where the page is composed by more boxes like in fig. 1. In this scenario a designer has to compose a web page considering a variety of options such as different layouts of the boxes that compose the page, size of the boxes, available products in promotion, background colors, presentation phrases and font types.

Is know that this process of combinatorial optimization is too difficult for a human, also if the designer employs his own skills in order to design what he considers the more user friendly combination. The designer has his model of the target customer but this could not be right at all, certainly not right as much as a model derived from a large analysis of the user behavior can be.

The only way for performing such a large analysis is to publish the page on the web and consider user reactions. Managing this interactive process for a large number of users is nearly impossible for a human, on the other hand an automatic optimization [13] of the content presentation can exploit the feedback of a large number of online users. Some applications [1, 13] has been proposed which try to select the optimal presentation using a voting mechanism (i.e. user feedback, such as number of clicks) among a fixed set of candidate ones, e.g. a set of candidate home pages, or by tuning some features parameters by randomly generating candidates to vote [2]. The limit in the first case is that the optimal solution could not be in the fixed set of candidates, while a purely random strategy can hardly find an optimal solution because of combinatorial explosion, especially in presence of features with nearly continuous values, such as colors or box sizes.

Recently particle swarm optimization (PSO) [3, 4] has emerged as an effective strategy for a variety of optimization problems. PSO uses the algorithmic metaphor of the dynamic of swarm behaviors in order to coordinate a set of particles, i.e. computational units, which move thru a given domain space. PSO combines cognitive and social strategies to focus the search of the swarm toward the most promising area.

The idea of our approach is to exploit the collective user feedback, as a swarm, in order to adapt and optimize the presentation features of web content with respect to the online audience. We devise a strategy based on PSO for generating candidates presentation instances which eventually converge to the optimal content presentation. A single particle submit a candidate presentation to a set of online users which provide a (implicit) feedback on it. Experiments show that the PSO strategy for web content presentation is effective and converges very fast, minimizing the number of sampled candidates (i.e. minimizing the number of non optimal presentations delivered), in the majority of cases. Unfortunately sometimes the convergence speed is not so high as in the other cases, however a solution to this problem was found and it consists in a procedure (to add to the standard PSO schema) of convergence speed recognition that restart the algorithm if the convergence speed goes below a certain threshold.

To the best of our knowledge this is the first proposal of application of PSO to the online optimization of content presentation, i.e. using an interactive approach where the user feedback is used as an objective function. Moreover an hybrid approach is proposed which combines continuous and discrete dimension in PSO (also if the discrete PSO has not been much addressed in the literature) other than the improvement to the PSO schema due to the restart procedure.

Other evolutionary approaches to adaptive web content selection has been proposed in the field of web newspapers [5, 6] and multimedia information retrieval [7]. In [6] newspapers layouts are adapted to a user web browser configuration by a PSO offline approach.

II. PSO SCHEMA AND PECULIARITIES

PSO has been introduced by Kennedy and Eberhart [3]. The metaphor got its inspiration from particles models of objects and simulation of collective behavior of flocks of birds. In PSO, a swarm is composed of a set of particles $P=\{p_1,p_2,\ldots,p_k\}$. The position of a particle corresponds to a candidate solution of a given optimization problem represented by an objective function $f\colon \Theta\rightarrow\Re$, with $\Theta\subseteq\Re^n$ (the set of feasible solutions), to be maximized (or minimized).

At any step t, each particle p_i has associated a position $x_{i,t}$, a velocity $v_{i,t}$, where position and velocity are n-dimensional vectors, and $b_{i,t}$ the particle personal best, i.e. the best position of p_i has ever visited until time step t. Moreover, particles are interconnected in a network and can communicate only with their neighbors l_i; in this way each particle can maintains the best position ever found among his l_i's neighbors denoted by $l_{i,t}$.

Each particle in the swarm moves according to its velocity. Position is updated by the vector expression

$$x_{i,t+1} = x_{i,t} + v_{i,t+1} \tag{1}$$

while velocity is updated by

$$v_{i,t+1} = \omega\, v_{i,t} + \varphi_1\, \beta_{1,t}\, (b_{i,t} - x_{i,t}) + \varphi_2\, \beta_{2,t}\, (l_{i,t} - x_{i,t}) \tag{2}$$

where the weights respectively represent the inertia ω, the acceleration factors φ_1, φ_2 and the random factors $\beta_{1,t}$, $\beta_{2,t}$ which are distributed in [0,1]. The contribution $(b_{i,t} - x_{i,t})$, the distance from the personal best, has been interpreted as a cognitive component, while $(l_{i,t} - x_{i,t})$ is a social component.

A number of variations to PSO has been proposed for velocity updating or other aspects. A very common one assumes that particles are connected by a complete network and in this case $l_{i,t}$ are substituted by a global l which can be maintained more efficiently. This simple variation is the one used in our approach.

Discrete PSO has been proposed since [8] and more recently [9], these methods emphasize the randomized contribution in order to obtain explorative behavior of the swarm in combinatorial search spaces. As pointed out in [10] and [3] PSO seems to benefit from the local monotony of objective function in continuous search spaces, but the same property does not hold in the discrete space generated by combinatorial problems. In other words, in a contour of a particle position, in a n-dimensional space, objective function is pleasant that is continuous, while in general this is not true for a discrete problem. On the other hand the distinction between continuous and discrete search space is not so sharp, since there are discrete spaces where the elements of the discrete domain can be ordered according to some notion of distance. Although this notion of "distance" does not have all the properties of a metric space but an approximation of continuity properties of objective function f can hold. Suppose for instance that f depends from different parameters such as temperature, and let the search space has a dimension T containing a finite set of ordered elements, say D_T=*{very_cold, cold, cool, mild, warm, very_warm, hot, very_hot}*, and suppose that the best value so far of f has been found in position T=*very_warm* then it is possible, and likely, to define an appropriate discretization of velocity and position update to make the other particles moving from their positions toward *very_warm* on dimension T.

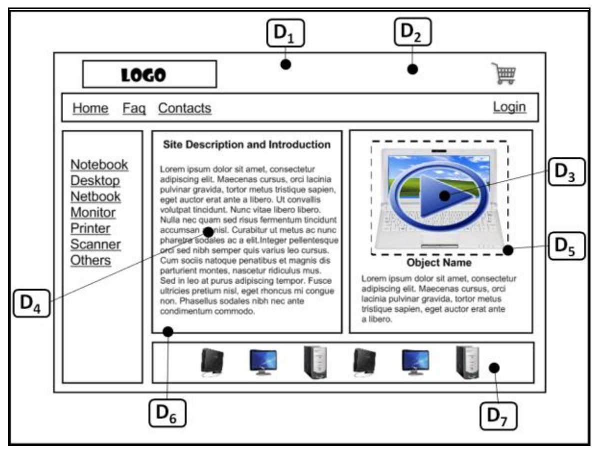

Figure 1. Content presentation structure and domains

III. ONLINE PSO SCHEMA AND MODEL FOR CONTENT PRESENTATION

A. Model of the Search Space

The search space consists of admissible content presentations.

The content presentation search space is described by a feature vector $C=[c_1,\ldots,c_n]$ with $c_i\in D_i$, where D_i are possible alternatives provided by the content presentation designer.

In content presentation problems, the domains D_i are, in general, a mix of continuous, discretized and pure combinatorial domains. Let consider for example the e-commerce home page scenario where a designer is proposing an home page for an online computer shop. The page consists of some boxes that can be disposed in different ways, a background which a uniform color, a main product in promotion with a video review, a sliding bar with the pictures of the other products in promotion, a site/company description and others (see fig. 1). The designer has selected

5 different layouts for the boxes (D_1); the background color of each box is fixed to be 5% more dark than that of the page but this has to be decided in a shade of grey (D_2); he is undecided among 7 different video reviews of the main product in promotion, 4 with audio and 3 without (D_3); he has decided the font types for the description text but not the font size which has to be among a range from 8 to 14 points (D_4); moreover the size of the main product video and of the description box can also vary in given ranges (D_5, D_6) and the number of products to show in the sliding bar has to be decided among a range from 5 to 15 (D_7). Summarizing the home page search space is $D=D_1\times D_2\times D_3\times D_4\times D_5\times D_6\times D_7$ where D_2, D_5, D_6, are continuous domains, D_1 is purely combinatorial, D_4 and D_7 are discretized and D_3 is discretized internally to each group of videos that are ordered internally for increasing duration time.

B. Online PSO Schema

The particle swarm algorithm proposed here uses a fully connected particle swarm.

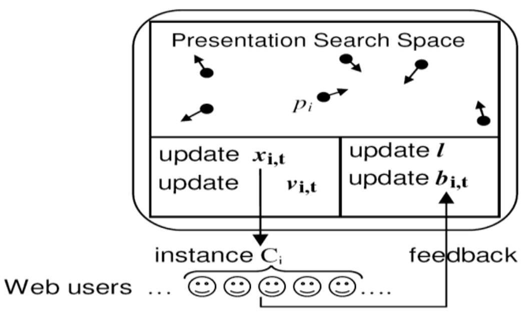

Figure 2. Online PSO for Web Content Presentation

At each iteration each particle generates a new candidate presentation configuration by moving to a new position $x_{i,t}$ in the search space D. The evaluation of the objective function $f(x_{i,t})$ is realized by submitting the candidate presentation to web users and measuring their feedback.

The feedback is used in order to determine the personal best, absolute best and in order to perform velocity update. The algorithm aims at maximizing the feedback function. A scheme of the online algorithm is shown in fig. 2.

The set P of particles is initially distributed in a random way in the search space. If the content designer has his own preferred or best candidate it is directly assigned to one particle. Personal best and global best are initially assigned to zero for all particles (i.e. no feedback observed).

Velocity updating has an important role in the proposed algorithm. Since it is supposed to have an hybrid continuous/discrete features space, different update functions are used for different classes of dimensional domains. The purpose is to exploit the local continuity for continuous and discretized domains and emphasize the exploration for pure combinatorial domains.

In the following is described the update phase of velocity and position which varies, as mentioned above, depending on the domain to which the feature belongs.

1) Continuous Feature

Position and velocity of continuous features, such as picture size, are updated according to the classical updating functions (1) and (2). Out of bound exceptions are managed by randomly restarting the particle dimension.

2) Ordered Discretized Feature

Discretized features are content features for which an order, or a similar notion, exists (i.e. domains D_3, D_4 and D_7 of the computer shop home page example).

The elements of an ordered discretized domain D_j are mapped into the integers $[0,1,2,\ldots|D_j|-1]$ and velocity/position updating is done as an integral approximation of a continuous position and velocity in the integral interval $[0, |D_j|-1]$.

$$x_{i,j,t+1} = \begin{cases} \lfloor x_{i,j,t} + v_{i,j,t+1} \rfloor & if \lfloor x_{i,j,t} + v_{i,j,t+1} \rfloor \in [0, |D_j|-1] \\ |D_j|-1 & if \lfloor x_{i,j,t} + v_{i,j,t+1} \rfloor > |D_j|-1 \\ 0 & if \lfloor x_{i,j,t} + v_{i,j,t+1} \rfloor < 0 \end{cases}$$

It is worth noticing that we do not use $Z_{|D_j|}$ arithmetic which has the unlikely property of "wrapping" on the interval extremes, instead, a "bouncing back" approach has been preferred since it seems to better preserve the local continuity of f.

When the particle goes out of the discrete bounds, its velocity is reset to a magnitude of 1 with direction towards inside the interval.

Let $v_{new} = \lfloor \omega\, v_{i,j,t} + \varphi_1 \beta_{1,t} (b_{i,j,t} - x_{i,j,t}) + \varphi_2 \beta_{2,t} (l - x_{i,j,t}) \rfloor$ then

$$v_{i,j,t+1} = \begin{cases} v_{new} & if \quad x_{i,j,t+1} \in (0, |D_j|-1) \\ 1 & if \quad x_{i,j,t+1} = 0 \\ -1 & if \quad x_{i,j,t+1} = |D_j|-1 \end{cases}$$

The experiments have shown that for very small discrete domain size, ω is usually 1 and the parameters φ_i should be less than 0.2.

3) Pure Combinatorial Feature

In the case of pure combinatorial features (i.e. alternative layouts), where an ordering is not defined, we use a randomized approach which guarantees domain exploration. Position is updated like for discretized domain while velocity is computed as

$$v_{i,j,t+1} = \begin{cases} \lfloor \beta_{3,t} \rfloor & if \ \lfloor \beta_{1,t} \rfloor = \lfloor \beta_{2,t} \rfloor = 1 \ or \ \lfloor \beta_{1,t} \rfloor = \lfloor \beta_{2,t} \rfloor = 0 \\ b_{i,j,t} - x_{i,j,t} & if \ \lfloor \beta_{1,t} \rfloor = 1 \ and \ \lfloor \beta_{2,t} \rfloor = 0 \\ l - x_{i,j,t} & if \ \lfloor \beta_{1,t} \rfloor = 0 \ and \ \lfloor \beta_{2,t} \rfloor = 1 \end{cases}$$

where $\beta_{3,t}$ is uniformly distributed in $[0,|D_j|-1]$, $\beta_{1,t}$ and $\beta_{2,t}$ are uniformly distributed in $[0,2)$.

C. Restart Procedure

This variation to the standard PSO schema is mainly due to these sporadic cases that show a convergence to a not good level of fitness (in a reasonable amount of time).

Mainly the procedure consists of a convergence speed recognition technique that restart the PSO algorithm if the convergence speed goes below a certain threshold. That is, if the best fitness so far (i.e. l for the fully connected network of particle used in our approach) is better of the best fitness k iterations ago of only a fixed quantity ε (or less) then the particles history is reset and the algorithm restart from the begin storing somewhere the best fitness value reached.

Note that this technique improves the convergence speed to a good fitness value for those bad and sporadic instances of the problem but it don't slow the convergence in the other cases, that is in these instances the "gain" of fitness in the last iterations will never be below the fixed threshold.

D. Online particle feedback evaluation

In order to perform a correct evaluation, each $f(x_{i,t})$ should be obtained showing the $x_{i,t}$ page to the same number of users, that is the sample size s must be constant. The cost of n iterations with $|P|$ particles is then the number $s \cdot n \cdot |P|$ of contacted users. In other words the flow of users is divided into n sets u_i, i=1,…,n, each of the same size s; an u_i represents the users assigned to evaluate f on the candidate solution currently found by particle p_i. The underlying assumption is that in all the user sample sets, f has an homogeneous behavior, i.e. the user are uniformly distributed in the n sets. Typical web marketing performance measures can be used, such as *#clicks / #impression*, *average_time_of_permanence*, *#finalized_orders / #impression* or a combination of these.

IV. Experiments

Experiments for the e-commerce home page scenario with hybrid features have been held using the hidden values technique developed in [5].

Optimal page features configurations are randomly generated and the simulated feedback function f of a candidate solution is computed as the euclidean distance (that is 0 or 1 for a pure combinatorial feature), normalized with the maximum distance admissible, between the candidate solution and the optimal solution randomly generated at the start of the experiment. Note that the optimal solution randomly generated is invisible to the PSO algorithm which only access to the evaluation function f.

First we have conduct some experiments (using a mixed space of continuous, discretized and pure combinatorial features) in order to determine the best number of particles that make up the population of PSO. As you can view from the chart in fig. 3 (where are compared the fitness value of different populations relatively to the number of fitness function evaluations) this parameter has a very little influence on our online PSO technique, hence, in the other experiments we have used a population of 20 particles (like suggested in [4]).

For each randomly generated optimal configuration the online PSO has been run for 200 iterations and in the chart below is reported the fitness value averaged over 200 random optimal configurations. Moreover artificial cases have been generated for comparison of different features mix: content presentation with only continuous features, discretized features and pure combinatorial features.

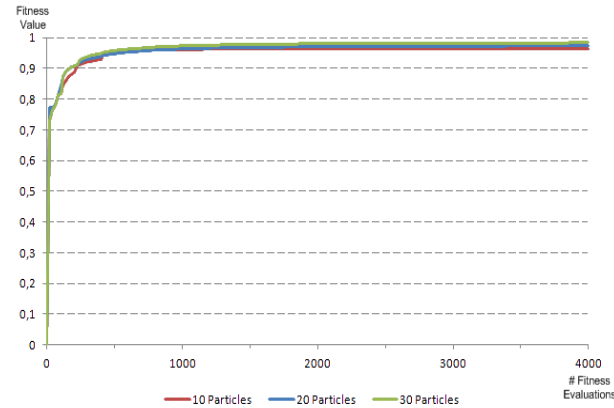

Figure 3. Population sizes comparison chart

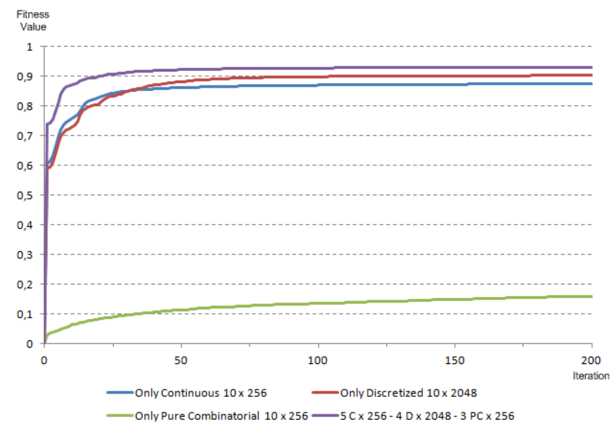

Figure 4. Performance chart without the restart procedure

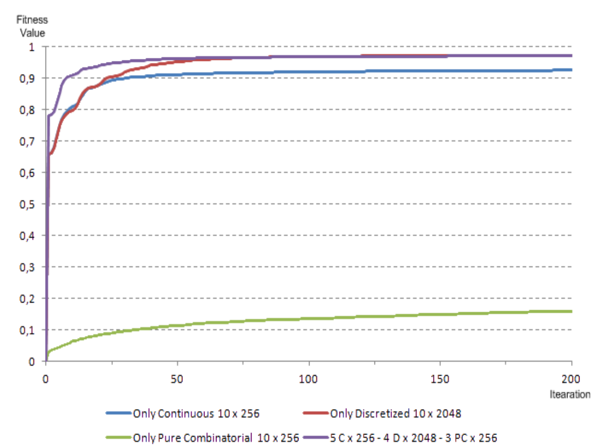

Figure 5. Performance chart with the restart procedure

Results show as expected that pure combinatorial problem domain easily become intractable while the best performances are obtained with continuous and discretized features (see fig. 4 and 5).

A justification to the fact that the speed of convergence of continuous features is slightly lower compared to that of the discretized features is that the cardinality of each continuous feature is infinitely larger than the cardinality of every discretized feature (this is the same difference that exists between a numeric real interval and a numeric integer interval); hence the candidate solutions for a continuous feature are infinitely more than the candidate solutions for a discretized feature.

The benefits of the restart procedure can be viewed comparing the chart in fig. 4 (without restart procedure) with that in fig. 5 (with restart procedure); in this case, for the experiments, we have used as parameters a k of 40 iterations and an ε of 0.02.

The mixed configuration used in the experiments is composed from 5 continuous features (with an interval of width 256), 4 discretized features (with a cardinality of 2^{10}) and 3 pure combinatorial features (with a cardinality of 2^8).

We are conduct experiments also for another simple variation of the online PSO schema suggested above, however some experiments conducted in this direction have not show an improvement of performance. This variation uses different weights in the computation of the fitness for any different feature in the search space. This was done specially to mitigate the influence of the pure combinatorial features over the others; but we could see that in a practical search spaces there are not too pure combinatorial features and with a little number of these kind of features there is no need of weights.

Although more experiments and parameters tuning are needed, the technique seem to be viable for online optimization of content presentation when the number of features is relatively small such as in many web content scenario.

V. Conclusions

A web marketing optimization model based on online PSO [3] has been presented. The method search the optimal configuration in the hybrid feature search space of web marketing content presentation. The online approach is based on collective user feedback [5] in order to guide the particle swarm toward the selection of the optimal presentation. Different kind of velocity updates are used in the hybrid multi-dimensional domains in order to exploit the local continuity of the objective function. Moreover a procedure of auto regulation of the convergence speed is used to improve performances when they are not so good.

It is worth noticing that the collective user behavior technique can be used to optimize in the same framework the web content presentation and some typical web marketing parameters such as adsense keywords to associate to marketing campaigns.

References

[1] M. Bolin, M. Webber, P. Rha, T. Wilson, R.C. Miller. “Automation and customization of rendered web pages”. In Proc. 18th ACM Symp. on User interface Software and Technology, ACM Press, New York, USA 2005, pp. 163-172.

[2] K. Marriott, B. Meyer, L. Tardif. “Fast and efficient client-side adaptivity for SVG”. In Proc.11th Int.Conf.onWorld Wide Web, Hawaii, USA, May 07, 2002.

[3] J. Kennedy, R. Eberhart. “Particle swarm optimization”. In Proc. of IEEE Conf. on Neural Networks, IEEE Press, 1995, pp. 1942-1948.

[4] R. Poli, J. Kennedy, T. Blackwell. “Particle swarm optimization. An overview.”. Swarm Intelligence, 1(1): 33-57, 2007.

[5] A. Milani. “Online genetic algorithms”, in International Journal of Information Theories and Applications, n.1 Vol.11, pp.20-28, (2004), ISSN 1310-0513.

[6] J.H. Lin, C.K. Wang, C.H. Lee (Taiwan). “Particle swarm optimization for web newspaper layout problem” in J.T. Yao (Ed) Web Technologies, Applications, and Services – 2006 pp. 524-026 ISBN0-88986-575-2.

[7] C. Leung, A. Chan. “Community adaptive search engines” in Int.J.of Advanced Intelligence Paradigms(IJAIP), Special Issue on Intelligent Techniques for Personalization and Recommendation, Inderscience, 2008, ISSN 1755-0386.

[8] J. Kennedy, R. Eberhart. “A discrete binary version of the particle swarm algorithm”. In Proc. of the IEEE Conf. on Systems, Man, Cybernetics, IEEE Press, 1997, pp. 4104-4108.

[9] X.H. Zhi, X.L. Xing, Q.X. Wang, Zhang. “A discrete PSO method for generalized TSP problem”, 2004. Proceedings of 2004 Int. Conference on Machine Learning and Cybernetics Vol. 4, 26-29 Aug. 2004, pp. 2378-2383.

[10] R. Poli. “Analysis of the publications on the applications of particle swarm optimisation”, J. of Artificial Evolution and Applications, Article ID 685175, 10 pages, 2008. ISSN 1755-0386.

[11] W.S. Chan, A. Milani, C.H.C. Leung, J. Liu. “An architectural paradigm for collaborative semantic indexing of multimedia data objects”, LNCS in Springer Verlag.

[12] A. Bunt, G. Carenini, C . Conati. “Adaptive content presentation for the web”, in P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), The Adaptive Web. Berlin: Springer, (2007), pp. 409-432.

[13] A. Jameson. “Adaptive interfaces and agents”, in Jacko, J., Sears, A. (eds.): Human-Computer Interaction Handbook, Erlbaum, Mahwah, NJ (2003), pp. 305-330.