Improving the Analysis of Context-Aware Information via Marker-Based Stigmergy and Differential Evolution

Mario G.C.A. Cimino(✉), Alessandro Lazzeri, and Gigliola Vaglini

Department of Information Engineering, University of Pisa, Pisa, Italy
{mario.cimino,gigliola.vaglini}@unipi.it,
alessandro.lazzeri@for.unipi.it

Abstract. We use the marker-based stigmergy, a mechanism that mediates animal-animal interactions, to perform context-aware information aggregation. In contrast with conventional knowledge-based models of aggregation, our model is data-driven and based on self-organization of information. This means that a functional structure called track appears and stays spontaneous at runtime when local dynamism in data occurs. The track is then processed by using similarity between current and reference tracks. Subsequently, the similarity value is handled by domain-dependent analytics, to discover meaningful events. Given the changeability of human-centered scenarios, the overall process is also adaptive, thanks to parametric optimization performed via differential evolution. The paper illustrates the proposed approach and discusses its characteristics through two real-world case studies.

Keywords: Context-aware information · Marker-based stigmergy · Optimization · Differential evolution

1 Introduction and Motivation

Context-awareness is a computing paradigm by which software systems can sense the user's context in order to provide personalized services. This paradigm relies on the *context*, that is, all information helping to understand what is happening in the user's physical or logical environment. Context-aware information can be supplied through different channels: data repositories, web applications, mobile applications, embedded systems, and so on [1]. To properly support service personalization, context-aware information should be adequately aggregated so as to detect human-centric events in a number of domains: financial transactions, health care needs, traffic jam, territorial emergency, and so on [2].

In the literature of context-awareness, at the core of aggregation of human-centric data is the construction of two possible types of model: (i) knowledge-based models, explicitly designed at the business level in terms of logical or mathematical rules, determined by a domain expert; (ii) data-driven models, i.e., systems that can learn from prototypical data via machine learning or statistical algorithm. Nevertheless, modeling and reusing application contexts remains a difficult task. An important lesson learned is that the algorithms performing the parametric data aggregation must use a limited number of states, be highly adaptable and handle variability [3,4].

Generally speaking, knowledge-based models belong to the *cognitivist* paradigm [5]. In this paradigm, the system is a descriptive product of a human designer, whose knowledge has to be explicitly formulated for a representational system of symbolic information processing. It is well known that knowledge-based systems are highly context-dependent, neither scalable nor manageable. With respect to knowledge-based models, data-driven models are more robust in the face of noisy and unexpected inputs, allowing broader coverage and being more adaptive. The data-driven approach discussed in this paper takes inspiration from the *emergent* paradigm [5], in which context information is augmented with locally encapsulated structure and behavior. Emergent paradigms are based on the principle of self-organization of data, which means that a functional structure appears and stays spontaneous at runtime when local dynamism in data occurs [6].

In this paper we propose to use the principles of the marker-based stigmergy to perform context-aware information aggregation. In biology, stigmergy is a class of mechanisms that mediate animal-animal interactions. It consists of indirect communication between individuals of an insect society by local modifications induced by these insects on their environment. Social insect colonies employ chemical markers (pheromones) that the insects deposit on the ground in specific situations. Pheromone concentrations in the environment disperse in space and evaporate over time, because pheromones are highly volatile substances. Multiple deposits at the same location aggregate in strength. Members of the colony who perceive pheromones of a particular flavor may change their behavior.

In computer science, marker-based stigmergy occurs when marks are left in an environment to enable self-coordination [7]. Marker-based stigmergy can be employed as a powerful computing paradigm exploiting both spatial and temporal dynamics, because it intrinsically embodies the time domain. Moreover, the mapping provided is not explicitly modeled at design-time and then not directly interpretable. This provides a kind of information blurring of the human data, and can be exploited to solve privacy issues.

In this work the main goal of data aggregation is to distinguish different spatio-temporal patterns occurring over time. For this purpose, we use stigmergic tracks for assessing similarity between context-aware data. Similarity is computed between a reference and a current track, and over different time periods, in order to measure the differences. Since context-data source is application-dependent, we have included an adaptive scheme on the marking and detection sub-processes. The setting of different applications consisting in different parameterizations can be automatically performed via a biologically-inspired optimization algorithm.

More specifically, the system architecture proposed in this paper is made of four subsystems: (i) the *marking* subsystem takes context information samples and releases marks in a computational environment; here, marks interact with each other at micro level generating a collective mark distribution. Collective mark can be considered as a short-term and a short-size memory which abstracts the complexity and the variability in the information sources; (ii) the *perception* subsystem compares the collective mark with a reference mark by similarity function; (iii) the *detection* subsystem connects similarity to specific application domain analytics; (iv) finally, the *adaptation* subsystem consists in the parametric optimization of the other subsystems. We used differential evolution among the other optimization methods [8].

The paper is structured as follows. Section 2 details on the first three subsystems, whereas Section 3 covers the adaptation subsystem. Both sections are based on two real-world pilot case studies. Finally, Section 4 draws some conclusions.

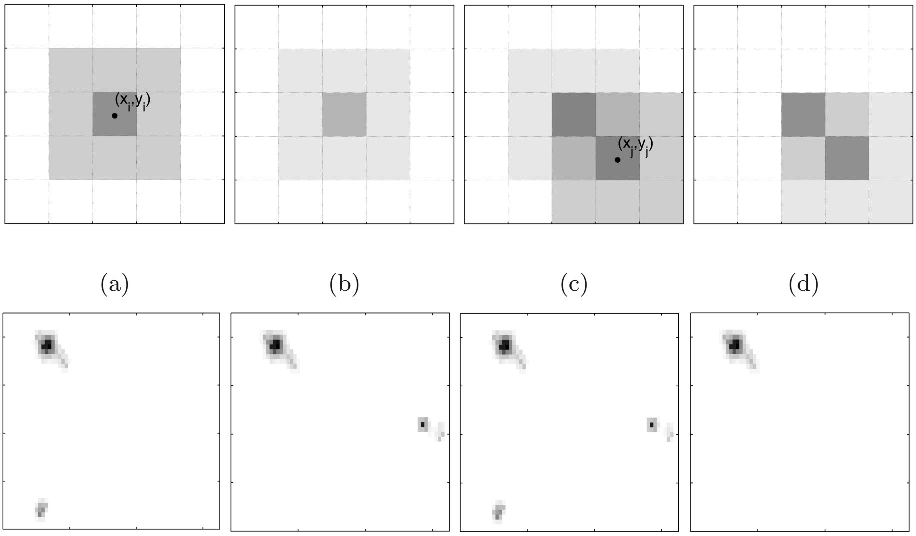
2 Processing with Stigmergy: A Three-Level Architecture

This section is focused on the marking, perception, and detection subsystems, described by considering a pilot real-world case study in the field of *ambient assisted living* (AAL): to monitor elderly people living alone in their own homes with the purpose of detecting possible disease situations. In the pilot case study, the context-aware input information is the x, y position of the elderly at home, periodically sampled, whereas the binary output is the detection of unusual behavior, with respect to a reference behavior sampled in a healthy period. The case study of the vendor rating is also presented.

2.1 The Marking Subsystem

The marking subsystem periodically takes as an input the position of the user at home and releases a mark in a computer-simulated spatial environment, thus allowing the accumulation of marks. A mark has four attributes: position (x, y), maximum intensity I_{MAX}, width ε, and evaporation θ. Fig. 1a-d shows some mark sample of the pilot scenario. The position of the elderly is represented by a dot, in Fig. 1a and Fig. 1c.

The levels of mark intensity are represented by different gray gradations: the darker the gradation is, the higher the intensity of the mark. In Fig. 1a the highest intensity of the mark I_{MAX} is in the middle, which corresponds to the position of the person where the mark is left. Mark intensity proportionally decreases with the number of squares from the position of the person, reaching its minimum at distance ε. Further, mark intensity has a temporal decay, i.e., a percentage θ of decrease after a step of time (tick). Hence, an isolated mark after a certain time tends to disappear, as shown in Fig. 1b sampled after a tick with respect to Fig. 1a. The time that a mark takes to disappear is longer than the period used by the marking subsystem to release a new mark. Hence, if the user is still in a specific position, new marks at the end of each period will superimpose on the old marks, thus increasing the intensity up to a stationary level. If the person moves to other locations, consecutive marks will be partially superimposed and intensities will decrease with the passage of time without being reinforced. Fig. 1c shows two consecutive and overlapping marks, and Fig. 1d shows the same track after a step of evaporation. The stigmergic track can then be considered as a short-term and a short-size action memory. The marking subsystem allows capturing a coarse spatiotemporal structure in the domain space, which hides the complexity and the variability in data.



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 1. (a) example of a two-dimensional mark; (b) the mark after a step of evaporation; (c), the aggregation of two consecutive marks; (d) the aggregated marks after a step of evaporation; (e) aggregated mark generated by an elderly moving in his apartment on a Friday at 19:20; (f) aggregated mark on the next Friday at 19:20; (g) the union of the marks of Fig. 1e and Fig. 1f; (h) the intersection of the marks of Fig. 1e and Fig. 1f.

2.2 The Perception Subsystem

At the second level there is the perception subsystem, consisting in the sensing of the track accumulated in the environment at the macro-level. Here, we take advantage of stigmergy (computed at the first level) as a means of information aggregation of the spatiotemporal tracks. Indeed, the process of information aggregation is a vehicle of abstraction, leading to the emergence of high-level behavior. The perception subsystem performs a comparison, called similarity, which aims at sensing the variation of the current behavior with respect to what was judged a normal behavior.

More specifically, given an accumulated mark, i.e., a track, the perception subsystem performs a similarity computation between the current track, T_i, and a reference track, T_i^{REF}, at the i-th step. A reference track is generated offline, by averaging the marks collected during healthy periods. Indeed in the case study the objective is to detect unusual behavior, and reference tracks were created when the elderly was healthy, for each day of week. Thus, the similarity with the current track and the reference track, $S(T_i, T_i^{REF})$, in the same day of a week provides information about unusual behavior.

Fig. 1e shows a two-dimensional representation of the track generated by an elderly moving in his apartment on a Friday at 19:20. Fig. 1f shows the track on the next Friday at 19:20. Fig. 1g shows the union of the tracks of Fig. 1e and Fig. 1f, whereas Fig. 1h shows the intersection of the same tracks. In general, given two marks, their similarity is a real value calculated as the volume covered by their intersection divided by the total volume (the union of them). The lowest similarity is zero (tracks with no intersection), whereas the highest is one (identical tracks).

2.3 The Detection Subsystem

The detection subsystem enhances and discovers relevant variation of the current distribution through sharpening and domain-dependent analytics. For this purpose, to achieve a better distinction of the critical phenomena, the s-shaped activation function is applied to the similarity output. As an effect, at each tick values lower than a lower threshold α are further decreased, whereas values higher than an upper threshold β are further amplified, to evidence major dissimilarity.

Fig. 2a shows the similarity values between current and reference track, in a sampling period of 14 hours and 35 minutes (175 total ticks, 1 tick corresponding to 5 minutes). A similarity value close to 1 means that there are no behavior differences, while a similarity close to 0 means that there are significant modification in behavior. Here, two horizontal dotted lines are also shown, representing sample values of the lower (α=0.4) and the upper (β=0.8) thresholds of the s-shape. In Fig. 2b, the thick line represents the s-shaped similarity, whereas the thin line represents actual behavioral changes, annotated by a human observer who analyzed video tracks of the elderly. From Fig. 2a and Fig. 2b it is apparent that three actual behavioral changes occurred, but only the third one is detected by the system. To improve the system quality, the system parameters can be better adjusted. In Fig. 2c a quality indicator is shown, by using the third event of Fig. 2b. More specifically, let $\ddot{t} = [\underline{t}, \overline{t}]$ be the duration of an actual event, and $\ddot{\tau} = [\underline{\tau}, \overline{\tau}] = [\min\{i|S(T_i, T_i^{REF}) = 0\}, \max\{i|S(T_i, T_i^{REF}) = 0\}]$ be the duration of an event detected by the system. To assess the error between the actual and the detected event we compute the one-dimensional similarity between the two time intervals: $S(\ddot{t}, \ddot{\tau}) = (\ddot{t} \cap \ddot{\tau})/(\ddot{t} \cup \ddot{\tau})$. From the interval arithmetic: $S(\ddot{t}, \ddot{\tau}) = \max\{0, \min(\overline{\tau}, \overline{t}) - \max(\underline{\tau}, \underline{t})\}/\{\max(\overline{\tau}, \overline{t}) - \min(\underline{\tau}, \underline{t})\} = (\overline{t} - \underline{\tau})/(\overline{\tau} - \underline{t})$. To assess the global error, the average similarity is calculated considering each j-th event.

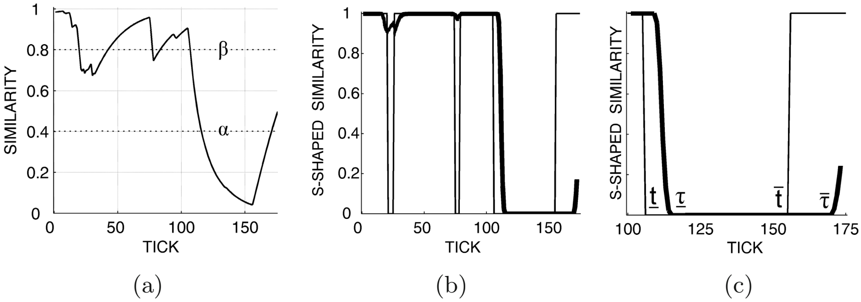


Fig. 2. (a) Similarity between current and reference tracks, in a sampling period of 14 hours and 35 minutes; (b) s-shaped similarity (thick line) with α=0.4 and β=0.8, against actual behavioral changes of the elderly (thin line); (c) start and end of an actual (thin line) and a detected (thick line) behavioral event

In order to show the generality of the approach, in the next section we briefly present a second real-world case study, concerning a vendor rating problem.

2.4 Application of the Approach to Another Case Study: Vendor Rating

Let us consider four manufacturing competitor firms, with the role of buyers with respect to a community of vendors. Context-aware information is provided by a community system for supplier relationship management, to carry out a *vendor rating* (VR). An important problem in the field is that, usually, a buyer is not willing to share the performance of his vendors, to keep a competitive advantage over its rivals. However, without information sharing each buyer can analyze only his subset of vendors. A solution to this problem is to use marker-based stigmergy for analyzing vendors context-aware information, so as to maximize its usability without violating its market value. Indeed, stigmergy preserves privacy since it controls the level of *perturbation* of information, which means that information is scrambled to be partially hidden but up to preserve its utility. Stigmergy allows masking plain information by replacing it with a mark, as a surrogate keeping some piece of the original information. The perturbation level can be controlled via mark's structural parameters. More specifically to increase the mark's width ε implies a higher uncertainty, whereas to decrease the evaporation θ implies a higher merging of past and new marks. A very large width ($\varepsilon \to \infty$) and a very small evaporation rate ($\theta \to 0$) may cause growing collective marks with no stationary level, because of a too expansive and long-term memory effect. A very small width ($\varepsilon \to 0$) and a very small evaporation rate ($\theta \to 0$) may cause the plain real values to appear for long time.

Fig. 3a-c shows three stigmergic perturbation levels applied to vendors' productivity values, calculated as output divided by labor, with increasing values of ε. We used information of publicly available dataset [9]. More specifically, in the marking subsystem, each k-th buyer locally produces a track T_k (represented as

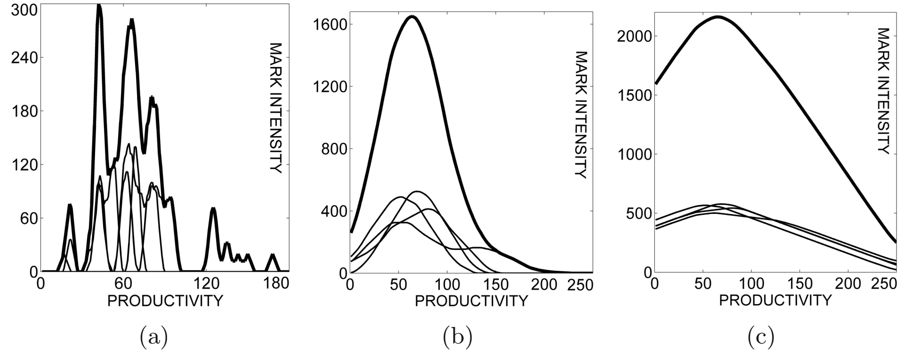


Fig. 3. Different stigmergic perturbation levels, in a community of four buyers, represented by individual track (thin line) and aggregated track (thick line). (a) low perturbation with ε=5: comparable but interpretable performance; (b) medium perturbation with ε=60: comparable and non-interpretable performance; (c) high perturbation with ε=200: non-comparable and non-interpretable performance.

a thin line in figure) by aggregating marks on the productivity of his vendors. In the perception subsystem, the tracks of the four buyers are aggregated and averaged online to create a reference track, T^{REF} (represented as a thick line in the figure), together with its average level, $\overline{T^{REF}}$, both shared between the buyers. The similarity value S_k between T_k and T^{REF} is then calculated. In the detection subsystem, such similarity is used as a performance indicator, to compare the buyer track with respect to the reference track. Moreover, to assess the utility of the information against its privacy, a quality indicator has been also defined as the product between ($\overline{T^{REF}}$) and the variance of S_k, to take into account two factors:

(a) *low perturbation* ($\varepsilon = 5$): a high variance of S_k and a low ($\overline{T^{REF}}$) makes the individual tracks easily interpretable from the aggregated track; as an example, Fig. 3a shows a bad scenario where an individual track in the interval [120,180] is not overlapped to other individual tracks and then it is transparent to the other buyers;

(b) *average perturbation* ($\varepsilon = 60$): an average variance of S_k and an average ($\overline{T^{REF}}$) makes the individual tracks totally overlapped, as shown in the good scenario of Fig. 3b;

(c) *high perturbation* ($\varepsilon = 200$): a low variance of S_k and a high ($\overline{T^{REF}}$) makes the buyers performance non-comparable, as shown in the bad scenario of Fig. 3c. In addition to ε, other structural parameters may also affect the perturbation level. As shown for both scenarios presented in this paper, to choose the parameters corresponding to the best quality of the performance indicators is crucial in the proposed approach. The next section is devoted to the adaptation subsystem, which traverses all levels of processing since it may affect all parameters to find the best setting.

3 Adapting the Stigmergic Process via a Cross-Level Subsystem

Table 1 summarizes, for each case study, the structural parameters set by a domain expert, and the corresponding quality metrics with their values. To adapt the structural parameters maximizing a quality metric is an optimization problem. The next subsection covers the design of the adaptation subsystem, which performs the optimization.

Table 1. Structural parameters set by a domain expert and quality metrics for each case study

Case	width (ε)	evap.(θ)	thr.(α, β)	Quality metric
AAL	10	0.9	0.4, 0.4	$Q = avg\{S(\ddot{t}_j, \ddot{\tau}_j)\}$=0.63
VR	60	0.9	0.0, 1.0	$Q = var\{S(T_k, T^{REF})\} \cdot avg\{T^{REF}\}$=4.57

3.1 The Adaptation Subsystem

Many optimization problems may be solved by *search* methods, i.e., procedures that look for a solution by trying out many attempts until a satisfactory result is obtained. Biologically inspired algorithms (BIAs) implement search mechanisms applicable to problems that cannot be efficiently solved using exact and analytical techniques [8]. Indeed, it is apparent from Table 1 that each case employs a different quality metrics. Then, an optimization method using a "black box" approach, i.e., which is not based on formal properties of the quality function, may be effective. Due to their random nature, BIAs can find near-optimal solutions rather the optimal solution.

BIAs optimize a problem by iteratively trying to improve a population of candidate solutions with regard to a given measure of quality, or *fitness*. Solutions are improved by means of stochastic transformation mechanisms inspired by biology, such as reproduction, mutation, recombination, selection, survival, swarm, movement, in an environment whose dynamics are represented by the quality measure.

Since the mid-sixties many BIAs have been proposed, and many efforts have also been devoted to compare them. In the last decade, most notably the following three classes of methods attracted attention: Genetic Algorithm (GA), Differential Evolution (DE), and Particle Swarm Optimization (PSO) [8]. A quantitative comparison of GA, DE, and PSO is beyond the scope of this paper. For the sake of brevity, an excerpt of their qualitative properties is summarized in Table 2 [8]. The interested reader is referred to the specialized literature for further details. It is apparent from Table 2 that DE is a simple and efficient adaptive scheme for global optimization. For this reason, it was selected to design the adaptation subsystem. Next subsection is devoted to DE and its different variants.

Table 2. An excerpt of the properties of the algorithms GA, PSO, and DE [8]

Property	GA	PSO	DE
Require ranking of solution	Yes	No	No
Influence of population size on solution time	Exponential	Linear	Linear
Influence of best solution on population	Medium	Most	Less
Average fitness cannot get worse	False	False	True
Tendency for premature convergence	Medium	High	Low
Density of search space	Less	More	More
Ability to reach good solution without local search	Less	More	More

3.2 The Differential Evolution

In DE algorithm, a solution is represented by a real D-dimensional vector, where D is the number of parameters to tune. DE starts with a population of N candidate solutions, injected or randomly generated. At each iteration and for each member (target) of the population, a mutant vector is created by mutation of randomly selected members and then a trial vector is created by crossover of mutant and target. Finally, the best fitting among trial and target replaces the target. More formally:

```
**DifferentialEvolution**()
  $P(0) \leftarrow$ InitializePopulation()
  $f \leftarrow$ ComputeFitness($P(0)$)
  $t \leftarrow 0$
  **while** $!stopCondition$ {
    **for each** $p(t) \in P(t)$ {
      $p' \leftarrow$ GenerateMutant($P(t)$, $p(t)$)
      $q \leftarrow$ Crossover($p(t)$, $p'$)
      **if** $f(q) < f(p(t))$
      **then** $p(t+1) \leftarrow q$
      **else** $p(t+1) \leftarrow p(t)$
    }
    $t \leftarrow t + 1$
    $f \leftarrow ComputeFitness(P(t))$
  }
```

In the literature, many variants of the DE algorithm have been designed, by combining different structure and parameterization of mutation and crossover operators [10]. Mutant vector is usually generated by combining three randomly selected vectors from the population excluding the target vector. More formally:

```
**GenerateMutant**($P$, $p$)
  $p_1, p_2, p_3 \leftarrow$ randomExtraction($P - p$)
  **return** $p_1 + F \cdot (p_2 - p_3)$
```

The scaling factor $F \in [0, 2]$ is a parameter of the DE algorithm. We used a commonly set value, i.e., $F = 0.8$.

There are different crossover methods. Results show that a competitive approach can be based on binomial crossover [11]. With binomial crossover, a component of the offspring is taken with probability CR from the mutant vector and with probability $1 - CR$ from the target vector. More formally:

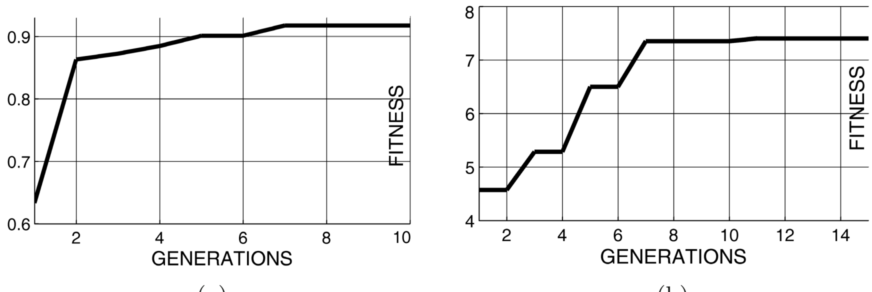
```
**BinomialCrossover**($p$, $q$)
  $k \leftarrow$ randomInteger(1, $n$)
  **for** $i = 1$ **to** $n$ {
    **if** randomReal(0,1)$< CR$ **or** $i = k$
    **then** $z_i \leftarrow p_i$
    **else** $z_i \leftarrow q_i$
  }
    **return** $z$
```

A small crossover probability leads to a vector that is more similar to the target vector while the opposite favors the mutant vector. We used a commonly set value, i.e., $CR = 0.7$, with the population size N equals to 15.

3.3 Experimental Studies

The aim of this section is to illustrate the possibilities offered by our approach, rather than to focus on a systematic optimization spectrum. For this purpose, we experimented the optimization offered by DE on both AAL and VR case studies. We used the parameters values found by domain experts as an initial (injected) solution, and the quality metrics already summarized in Table 1.

Fig. 4 shows the fitness versus number of generations for both cases. Here, it is apparent that the parametric optimization sensibly improved the initial setting, after a small number of generations (about 10) and with a very fast convergence: the quality metric has been highly improved, up to 44% and 62%, for AAL and VR, respectively. The parameters and the quality metrics (fitness) values



(a) (b)

Fig. 4. Fitness versus number of generations for two case studies: (a) AAL; (b) VR

Table 3. Best solution provided by the adaptation module for each case study

Case	width (ε)	evap.(θ)	thr.(α, β)	Quality metric
AAL	37	0.51	0.52, 0.52	0.91
VR	42	0.94	0.00, 1.00	7.40

provided at the end of the optimization processes are summarized in Table 3, to be easily compared with the values set by domain experts (Table 1). In the AAL case ε has been sensibly increased and θ has been considerably reduced, whereas in the VR case ε has been strongly reduced keeping θ about constant. To focus the analysis on the marking structure, α and β were constrained to be equal (AAL) or fixed to constant values (VR).

4 Conclusions

We have presented a novel approach to analyze context-aware information. The approach is based on representing the context datum as a mark, to enable self-organization between data. An architecture exploiting the mechanisms of the marker-based stigmergy have been designed and discussed on two real-world domains. An adaptation subsystem based on differential evolution has been also designed and experimented to enable a self-parameterization of the architecture. Experimental results show the effectiveness of the approach. However, to ensure high-quality design, the system should be cross-validated against more dynamic context data series. Indeed, one of the problems to solve when optimizing parameters is that optimization encompasses all available scenarios at once and may include different contexts, spread across the entire search space. This global tuning leads to increasing difficulties from the practical perspective, due to fitting different scaled spatiotemporal data. An alternative is local modeling, which requires an architecture based on sub-models that focus predominantly on some selected regions of the entire domain. An overall model is then formed by combining such local models. This modular layer may provide a topology offering a considerable level of flexibility, as the resulting sub-models can be highly diversified according to the distribution of the local data. For this reason, future work will be focused on using more dynamic context data series, to assess the fitting properties of the current system and to enable the design of a composite architecture.

References

1. Cimino, M.G.C.A., Lazzerini, B., Marcelloni, F., Ciaramella, A.: An Adaptive Rule-Based Approach for Managing Situation-Awareness. Expert Systems With Applications 39(12), 10796–10811 (2012)
2. Feng, L., Apers, P.M.G., Jonker, W.: Towards context-aware data management for ambient intelligence. In: Galindo, F., Takizawa, M., Traunmüller, R. (eds.) DEXA 2004. LNCS, vol. 3180, pp. 422–431. Springer, Heidelberg (2004)

3. Ciaramella, A., Cimino, M.G.C.A., Marcelloni, F., Straccia, U.: Combining Fuzzy Logic and Semantic Web to Enable Situation-Awareness in Service Recommendation. In: Bringas, P.G., Hameurlain, A., Quirchmayr, G. (eds.) DEXA 2010, Part I. LNCS, vol. 6261, pp. 31–45. Springer, Heidelberg (2010)
4. Ciaramella, A., Cimino, M.G.C.A., Lazzerini, B., Marcelloni, F.: A Situation-Aware Resource Recommender Based on Fuzzy and Semantic Web Rules. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems (IJUFKS) 18(4), 411–430 (2010)
5. Vernon, D., Giorgio, M., Giulio, S.: A survey of artificial cognitive systems: Implications for the autonomous development of mental capabilities in computational agents. IEEE Transactions on Evolutionary Computation 11(2), 151–180 (2007)
6. Avvenuti, M., Daniel, C., Cimino, M.G.C.A.: MARS, a Multi-Agent System for Assessing Rowers' Coordination via Motion-Based Stigmergy. Sensors 13(9), 12218–12243 (2013)
7. Van Dyke Parunak, H.: A survey of environments and mechanisms for human-human stigmergy. In: Weyns, D., Van Dyke Parunak, H., Michel, F. (eds.) E4MAS 2005. LNCS (LNAI), vol. 3830, pp. 163–186. Springer, Heidelberg (2006)
8. Kachitvichyanukul, V.: Comparison of three evolutionary algorithms: GA, PSO, and DE. Industrial Engineering & Management Systems 11(3), 215–223 (2012)
9. Bache, K., Lichman, M.: UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science(2013), http://archive.ics.uci.edu/ml
10. Mezura-Montes, E., Velázquez-Reyes, J., Coello Coello, A.: A comparative study of differential evolution variants for global optimization. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO), pp. 485–492. ACM (2006)
11. Zaharie, D.: A comparative analysis of crossover variants in differential evolution. In: Proceedings of IMCSIT 2007, pp. 171–181 (2007)