A Multi-Agent Memetic System for Human-based Knowledge Selection

Giovanni Acampora, *Member, IEEE,* Jose Manuel Cadenas,
Vincenzo Loia, *Senior Member, IEEE,* and Enrique Muñoz

***Abstract*—In these last decades, both industrial and academic organizations have used extensively different learning methods to improve humans' capabilities and, as consequence, their overall performance and competitiveness in the new economy context. However, the rapid change in modern knowledge, due to exponential growth of information sources, is complicating learners' activity. At the same time, new technologies offer, if used in a right way, a range of possibilities for the efficient design of learning scenarios. For that reason, novel approaches are necessary to obtain suitable learning solutions able to generate efficient, personalized and flexible learning experiences. From this point of view, Computational Intelligence methodologies can be exploited to provide efficient and intelligent tools able to analyze learner's needs and preferences and, consequently, personalize its knowledge acquirement. This paper reports an attempt to achieve these results by exploiting an ontological representation of learning environment and an adaptive memetic approach, integrated into a cooperative multi-agent framework. In particular, a collection of agents analyzes learner preferences and generate high quality learning presentations by executing, in a parallel way, different cooperating optimization strategies. This cooperation is performed by jointly exploiting data mining, via fuzzy decision trees, together with a decision making framework exploiting fuzzy methodologies.**

***Index Terms*—Adaptive Memetic Algorithms, Multi-Agent Systems, Data Mining, Fuzzy Logic, e-Learning**

I. Introduction

For many years technology enhanced learning [1] has been commonly implemented using information transfer paradigm, i.e., a teacher provides educational contents to be transferred to learners, and learners consume these contents in a passive way. Consequently, most e-learning solutions simply "digitize" this approach resulting in distance learning software platforms. These platforms compute *learning presentations* by taking into account a sole kind of input, the *educational resources*. This limitation, together with others like support for pedagogies, the contextualization of the learning experience or the engagement of the learner in activities, are considered the main barriers to e-learning adoption and trace the path for future developments.

In particular, the existing e-learning solutions show the following drawbacks:

1) they force learners to learn without taking into account their dispositions or preferences;
2) they constrain learners to learn and teachers to teach following a predefined approach;
3) they do not use the diverse tools provided by Web 2.0, that can improve learners' experience.

In order to face these weaknesses, modern e-learning solutions must represent a real evolution where the main efforts should be devoted to support the whole learning process, not only specific parts of the process (i.e., content management, resource delivery, etc.) and they should incorporate tools like Blogs (used to share ideas), Wikis (used as a way to construct knowledge in a collaborative way), Podcast (used to distribute multimedia files over the Internet) and other Web Sharing Applications (e.g., Flickr, YouTube, del.icio.us, etc.) in order to allow e-learning users to work, teach, learn, make business, etc.. The e-learning processes that satisfy these requirements, by coherently using aforementioned tools, belong to e-learning 2.0 scenarios [2]. Consequently, next generation Learning Management Systems (LMS) [3] should support new e-learning trends by means of suitable models and processes that dynamically and intelligently create e-learning experiences (i.e., a structured sequence of learning activities composed by contents and services able to facilitate learners to acquire a set of competences about a specific domain) adapted to learner expectations and objectives in the Web 2.0 environment.

This result can be achieved by defining personalized e-learning experiences able to maximize the understanding level of learners with respect to specific learning objectives[4]. Moreover a representation model to describe both educational domains and learners characteristics is required. In this scenario, ontological methodologies [5] [6] model and represent the e-learning contexts realizing a conceptualization of an educational domain by identifying its relevant subjects and organizing them by means of a fixed set of relations [7]. Analysis of ontological relationships enables a convenient way to bind subjects with learning activities in order to define personalized e-learning experiences. We name this binding problem as *Learning Presentation Problem*. A possible approach to face the learning presentation problem is to consider it as a modified version of the Uncapacitated Facility Location Problem (UFLP) [8] and solve it through a specific deterministic algorithm. This approach is convenient when we are considering a repository with a limited number of learning activities and learning paths with few subjects.

But when the execution environment is the Web 2.0 [9] with distributed repositories and the learning paths are made by numerous subjects we need to investigate other solutions.

G. Acampora and V. Loia are with the Department of Mathematics and Computer Science, University of Salerno, 84084, Fisciano, Salerno, Italy. E-mail: {gacampora, loia}@unisa.it

J. M. Cadenas and E. Muñoz are with the Department of Information and Communications Engineering, University of Murcia, 30071, Espinardo, Murcia, Spain. E-mail: {jcadenas, enriquemuba}@um.es

Manuscript received April 19, 2005; revised January 11, 2007.

In particular, we propose a multi-agent memetic algorithm aimed to solve optimization problems by deciding, depending on the instance being solved, how to combine and configure different evolutionary and local search metaheuristics [10] in order to derive high-quality learning experiences. In detail, the proposed algorithm computes two sequential steps dealing respectively with evolutionary and local methodologies. During the evolutionary step a collection of population based optimization methods cooperates in order to find a good experience that is forwarded to the second step where a collection of local search methods cooperate to improve it. The metaheuristics cooperation is performed by exchanging solutions among optimization methods in precise moments and under certain conditions that are controlled by a set of fuzzy rules. The fuzzy engine uses knowledge obtained by a preliminary machine learning process that analyzes the performances of different optimization methods applied to well-defined problem's instances. This methodology is very suitable for Learning Presentation Problem, as the information about the behavior of previous learners can help the algorithm to foresee the behavior of future ones, and thus obtain more accurate personalized e-learning experiences. As will be shown in the experimental results section, proposed strategy is capable of speeding up the convergence to high-quality personalized e-learning experiences.

The paper is organized as follows, in Section II we present some related work, in Section III a description of ontological e-learning environment is provided in order to define the Learning Presentation Problem that will be solved using our adaptive memetic algorithm presented in Section IV, whereas in Section V we test the efficiency and suitability of proposed optimization framework on a real learning context. Finally, in Section VI we present the conclusions.

II. Related Work

Today, e-learning solutions implement simple distance learning software platforms based on different kinds of technology specifications. In particular, the learning technology specifications are based on educational metadata: IEEE LOM[1] and ADL SCORM [11], for example, are the standards proposed for describing a piece of learning content annotated through metadata [12]. Metadata is supposed to enable reuse of these chunks by detailing the conditions of their initial deployment [13]. However, some authors [14], [15] pointed out that such an approach failed to elicit cognitive behaviours and therefore actual reuse. We believe that the use of ontologies in e-learning can overcome these drawbacks. In [16] some benefits of applying ontologies to e-learning are well explained. The authors assert that an ontology, formally and declaratively, represents the terminology of a specific domain, defining its essential knowledge. Ontologies are used to support semantic search, making possible to query multiple repositories and discover associations, between Learning Objects, that are not directly understandable. This is impossible or very complex with simple keyword-based or metadata-based search supported by the current standards.

[1]http://ltsc.ieee.org/wg12

Blended learning is considered a learning approach defined by the effective combination of different modes of delivery and models of teaching and styles of learning [17]. Personalized e-learning experiences represent a convenient way to complement face-to-face sessions within a whole blended learning experience. A personalized e-learning experience could be very important when used for assessing the knowledge acquired by each individual learner during a face-to-face learning session and offering, in case of negative results, personalized remedial works able to fill the identified knowledge gaps with the learning paths that best fits the needs, the cognitive state and the learning preferences of each individual learner. In the event that the personalized e-learning experience can be built, packaged and deployed with an automatic process, then the whole blended learning activity can become more effective and efficient [18].

Computational Intelligence techniques are meaningfully exploited for providing intelligence and personalization in e-learning environments [19], [20]. Buraga [21] proposes an agent-oriented extensible framework based on XML family for building a hypermedia e-learning system available on the world-wide web. It deploys mobile agents that can exchange information in a flexible way via XML-based documents. Angehrn et al. [22] suggest the use of K-InCA to provide a personalized e-learning system. K-InCA is an agent-based system designed to assist people in adopting new behaviors. The agents within the system examine users actions and maintain a "behavioral profile" reflecting the level of adoption of the desired behaviors. In order to recommend useful learning material to students, Andronico et al. [23] propose an e-learning system that suggests educational resources to students into a mobile learning platform that supports mobile learning processes via a multi-agent recommendation system [24], where the aim is to collect data about the user behavior and preferences and then suggest educational resources to them according to their profile.

Web mining represents an interesting technique for assisting learners during online learning process. Zaiane [25] suggests the use of web mining techniques to build an agent that could recommend on-line learning activities or shortcuts in a course web site based on learners access history to improve course material navigation as well as assist the online learning process. The automatic recommendation system takes into account profiles of on-line learners, their access history and the collective navigation patterns, and uses simple data mining techniques. According to the service-oriented integration for e-learning systems based on web services, Pankratius et al. [26] provide a related architecture and describes the extensions to support software agents. It focuses on using intelligent software agents for the distributed retrieval of educational content. In this architecture intelligent software agents can be used to acquire user specifications of learning content and search for matching Learning Objects on behalf of the user. The work utilizes the web services as a wrapper around Learning Objects.

Different from other proposals trying to exploit users' preferences to derive adaptive learning experiences, our approach introduces an important concept, the ontological representation

of learning environment. Starting from this representation, the proposed system defines an optimization problem whose solution represents the most suitable choice of learning resources that a given learner has to study in order to improve his skills.

III. The Ontological e-Learning Environment

The proposed e-Learning system is based on 1) a set of models able to represent the main entities involved in the process of teaching/learning and on 2) a set of methodologies, leveraging on such models, for the generation of individualized learning experiences with respect to learning objectives, pre-existing knowledge and learning preferences. The three models adopted by our solution, based on the *Learning Model* [27], are summarized below:

- the *domain model* represents the knowledge domain that is object of teaching by means of concepts, relations among concepts and teaching preferences connected with concepts;
- the *learner model* represents a learner including concepts that he knows as well as his learning preferences;
- the *learning activity model* represents a single learning object or service that may be used as a building block to generate a learning experience [28];
- the *unit of learning model* represents a whole learning experience personalized for a single learner and composed by a set of target concepts and a sequence of learning activities needed to learn the target concepts.

Our system uses these models to automate some of the phases of the teaching/learning process. In particular the teacher may initialize a unit of learning by setting target concepts and associate one or more learners to it (in self-directed learning, learners settle own target concepts and constraints by themselves). Then the system generates a personalised *learning path* (sequence of concepts to be taught) for each learner through a *learning path generation algorithm* and introduces placeholders for testing activities.

Once the learning path is available, the system selects a fragment of the learning path and efficiently generates the best *learning presentation* (sequence of learning activities) for each enrolled learner by applying our proposal of Multi-Agent Memetic Algorithm. The learner then undertakes the learning and testing activities of the learning presentation until its end. When the learner ends a presentation fragment, his learner model is updated on the basis of the results of testing activities and a new learning presentation fragment is generated (possibly including recovery activities for concepts that he did not understand).

A. The Domain Model

The domain model describes, in a machine-understandable way, the piece of the educational domain that is relevant for the e-learning experience we want to define, concretize and broadcast. The mechanism used is named ontology, i.e., an engineering artefact, constituted by a specific vocabulary used to describe a certain reality, plus a set of explicit assumptions regarding the intended meaning of the vocabulary words [29].

TABLE I
Example of Didactical Properties and Feasible Values.

Properties	**Feasible Values**
didactic method	deductive, inductive, etc.
activity type	text reading, video clip, simulation discussion with a peer, discussion with the teacher, etc.
interactivity level	high, medium, low

In our approach the ontology is composed by a set of concepts (representing the topics to be taught) and a set of relations between concepts (representing connections among topics). Such a structure can be formally represented as a $(n+1)$-tuple $G(C, R_1, \ldots, R_n)$ where C is the set of nodes representing domain concepts and each R_i is a set of arcs corresponding to the i^{th} kind of relation. Several kinds of relations are allowed. As an example we can consider a concept graph $G(C, BT, IRB, SO)$ with three relations BT, IRB and SO whose meaning is explained below (where a and b are two concepts of C):

- $BT(a, b)$ means that the concept *a belongs to b*, i.e., b can be understood if every a so that a belongs to b is already learned. (hierarchical relation);
- $IRB(a, b)$ means that the concept *a is required by b*, i.e., a necessary condition to study b is to have understood a (ordering relation);
- $SO(a, b)$ means that the *suggested order* between a and b is that a precedes b, i.e., to favour learning, it is desirable to study a before b (ordering relation).

Figure 1 shows a sample domain model in the didactics of artificial intelligence exploiting the relations defined above. In particular, this sample shows that in order to understand "logics" means to understand "formal systems", "propositional logic" and "first order logic" but, before approaching any of these topics it is necessary to have an "outline of set theory" first. Moreover, "formal systems" must be taught before both "propositional logics" and "first order logic" while it is desirable (but not compulsory) to teach "propositional logics" before "first order logic".

A set of *teaching preferences* may be added to the domain model to define feasible teaching strategies that may be applied for each available concept. Such preferences are represented as a function $TP(C \times Props \times PropVals) \rightarrow [0, 10]$ where $Props$ is the set of didactical properties and $PropVals$ is the set of feasible values for such properties. The Table I provides some (non exhaustive) example of didactical property and associated feasible values. It is worth noting that TP is defined only for couples of $Props$ and $PropVals$ elements belonging to the same row in Table I.

B. The Learner Model

The learner is the main actor of the whole learning process and it is represented with a cognitive state and a set of learning preferences [30]. The *cognitive state* represents the knowledge reached by a learner at a given time and it is represented as

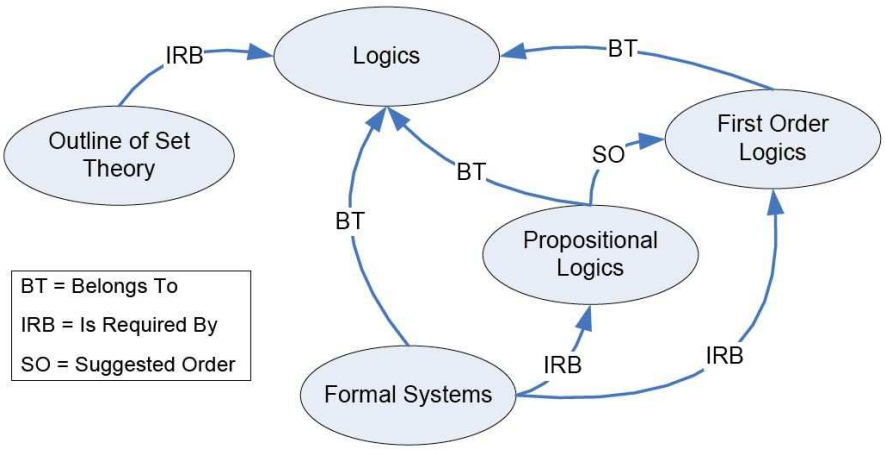

Fig. 1. A Sample Domain Model.

a function $CS(C) \rightarrow [0, 10]$ where C is the set of concepts of a given domain model. Given a concept c, $CS(c)$ indicates the degree of knowledge (or grade) reached by a given learner for c; the value 0 indicates *no knowledge*, whereas, the value 10 indicates *full knowledge*. If such grade is greater than a given "passing" threshold θ then c is considered as known, otherwise it is considered as unknown.

The *learning preferences* provide an evaluation of learning strategies that may be adopted for a given learner. They are represented as an application $LP(Props \times PropVals) \rightarrow [0, 10]$ where $Props$ and $PropVals$ are the same sets defined in Table I for teaching preferences, whereas, the value 0 and 10, represent, respectively, the lowest and highest level of learning preference. Differently from teaching preferences, learning preferences are not linked to a domain concept but refer to a specific learner. The cognitive state of any learner is initially void (i.e., $CS(c) = 0$ for any c included in a given domain model) and may be initialized on a teaching domain with a pre-test. Learning preferences may be initialized by the teacher or directly by learners through a questionnaire capable of evaluating learners styles and transform them in suitable values for learning preferences. Both parts of the learner model are automatically updated by the system during learning activities.

C. The Learning Activities Model

A *learning activity* must be performed by a learner to acquire one or more domain concepts. Activities may relate to learning objects (e.g., textual lessons, presentations, videoclips, podcasts, simulations, exercises, etc.) or learning services (e.g., virtual labs, wikis, folksonomies, forums, etc.). Our system uses learning activities as bricks to generate learning experiences. In order to be used in this way, a learning activity LO is described through the following elements:

- a set of *concepts* C_{LO} part of a given domain model, that are covered by in the learning activity;
- a set of *didactical properties* expressed as an application $DP_{LO}(property) = value$ representing learning strategies applied by the learning activity;
- a set of *cost properties* expressed as an application $CP_{LO}(property) = value$ that must be taken into account in the optimisation process connected with the *learning presentation generation algorithm.*

Didactical properties components have the same meaning with respect to teaching and learning preferences, i.e., property and value may assume values from a closed vocabulary (see table I). Differently from learning and teaching preferences, they are neither linked to a domain concept nor to a specific student but to a learning activity. Cost properties are couples that may be optionally associated to learning activities, whose properties may assume values from the closed vocabulary {price, duration} and whose values are positive real numbers representing, respectively the price of a single learning resource and its average duration in minutes.

D. The Unit of Learning Model

A unit of learning represents a sequence of learning activities needed for a learner in order to understand a set of target concepts in a given domain with respect to a set of defined cost constraints [31], [32]. It is composed by the following elements:

- a set of *target concepts* TC part of a domain model, that have to be mastered by a given learner in order to successfully accomplish the unit of learning;
- a set of *cost constraints* $CC(property) = value$ that must be taken into account in the optimisation process connected with the learning presentation generation algorithm;
- a *learning path* $LPath(c_1, \ldots, c_n)$, i.e., an ordered sequence of concepts that must be taught to a specific learner in order to let him master target concepts;
- a *learning presentation* $LPres(lo_1, \ldots, lo_m)$, i.e., an ordered sequence of learning activities that must be

presented to a specific learner in order to let him/her master the target concepts.

While target concepts are defined by the course teacher, the learning path and the learning presentation are created by the generation algorithms described below. Concerning cost constraints, the property may assume values from the closed vocabulary {price, duration}. Feasible values are positive real numbers representing, respectively the maximum total price and the maximum total duration of the unit of learning.

The generation of the learning path is the starting point to completely generate a unit of learning. Starting from a set of *target concepts* TC and from a *domain model*, a feasible learning path must be generated taking into account the concepts graph $G(C, BT, IRB, SO)$ part of the domain model (with $TC \subseteq C$).

The four steps of the learning path generation algorithm are summarized below:

- The *first step* builds the graph $G'(C, BT, IRB', SO')$ by propagating ordering relations downward the hierarchical relation. IRB' and SO' are initially set to IRB and SO respectively and then modified by applying the following rule: for each arc $ab \in IRB' \cup SO'$ substitute it with arcs ac for all $c \in C$ such that there exist a path from c to b on the arcs from BT.
- The *second step* builds the graph $G''(C', R)$ where C' is the subset of C including all concepts that must be taught according to TC, i.e., C' is composed by all nodes of G' from which there is an ordered path in $BT \cup IRB'$ to concepts in TC (including target concepts themselves). R is initially set to $BT \cup IRB' \cup SO'$ but all arcs referring to concepts external to C' are removed.
- The *third step* finds a linear ordering of nodes of G'' by using a depth-first search that visits the graph nodes along a path as deep as possible. The obtained list L will constitute a first approximation of the learning path.
- The fourth step generates the final learning path $LPath$ by deleting from L all non-atomic concepts with respect to the graph G, i.e., $LPath$ will include any concept of L apart concepts b so that $ab \in BT$ for some a. This ensures that only leaf concepts will be part of $LPath$.

Successively it is necessary to generate the learning presentation suitable for a specific learner based on a *learning path* $LPath$ that has to be covered, on a set of teaching preferences TP belonging to a *domain model*, on a cognitive state CS and a set of learning preferences LP both part of the *learner model* associated to the target learner, on a set of optional *cost constraints* CC and on a set of available *learning activities*. The learning presentation is generated by following the steps below:

- The *first step* is to select the sub-list L of $LPath$ that has to be converted in a presentation. L is the sequence of all the concepts of $LPath$ not already known by the learner (i.e., any concept a so that $CS(a) < \theta$). If L is empty then the algorithm ends because the learner already knows all concepts of the learning path.
- The *second step* is to define the best sequence of learning activities P, selected from available learning activities, covering L on the basis of TP, LP and CC. This sequence definition has been given the name of *Learning Presentation Problem (LPP)*.

LPP problem represents the core of this work, in fact, its solution defines the learning experiences personalization. Hereafter, LPP problem will be modeled by means of well known UFLP problem and, successively, it will be efficiently solved through an innovative memetic algorithm.

In order to model LPP by means of UFLP problem, first of all a measure of distance $d_{TP}(lo, c)$ between an activity lo and the set of preferences TP has to be defined with respect to a concept c. In a similar way a measure of distance $d_{LP}(lo)$ based on LP may be defined. A further measure $d(lo, c)$ shall be defined as a weighted sum of the two measures.
Once the measure of distance is defined the problem of selecting the best set of activities P covering concepts of L becomes a UFLP that may be outlined with the bi-partite graph in Fig. 2, where available activities are displayed on the left and concepts to be covered on the right.

There is an arrow between a learning object LO_i (considering the available activities related to learning objects) and a subject C_j only if the metadata instance of LO_i includes a semantic link to the subject C_j

For each couple (LO_i, C_j) linked in the bipartite graph there is an assigned value $d(i, j)$ representing the distance between LO_i and C_j. Short distances define good coverings. Furthermore, to each learning object LO_i, a value $p(i)$ representing the cost of the introduction of the learning object LO_i into the sequence of LOs delivered to the learner is associated . Assume that distances $d(i, j)$ are calculated applying a specific function that evaluates the matching between the metadata values of LO_i covering C_j and the learning preferences of the learner who requests the personalized e-learning experience. Then, P must be built as the smallest set of activities covering all concepts of L with the minimum sum of distances between activities and covered concepts. Now, consider m as the number of learning objects available and n the number of subjects in the Learning Path to be filled. Set y as a binary vector where each element y_i, $(i = 1, \ldots, m)$, assumes value 1 if you decide to use the learning object LO_i, 0 otherwise, and x as binary vector in which each entry x_{ij}, $(i = 1, \ldots, m$ and $j = 1, \ldots, n)$, assumes value 1 if subject C_j is covered by the learning object LO_i, 0 otherwise. Then, the linear programming model which formalizes the whole problem is described as follows:

$$\min \sum_{i=1}^{m} p(i) y_i + \sum_{i=1}^{m} \sum_{j=1}^{n} d(i, j) x_{ij} \tag{1}$$

subject to constraints

$$\begin{array}{lll} \sum_{j=1}^{n} x_{ij} = 1 & i = 1, \ldots, m & \\ x_{ij} \leq y_i & i = 1, \ldots, m & j = 1, \ldots, n \\ x_{ij} \in \{0, 1\} & i = 1, \ldots, m & j = 1, \ldots, n \\ y_i \in \{0, 1\} & i = 1, \ldots, m & \end{array} \tag{2}$$

The optimal solution of the LPP means the identification of the optimal set of learning objects that better match (minimal distance) the learner preferences. As any other optimization

problem, it can be solved by means of a deterministic algorithm. This approach is valid when we consider a repository with a limited number of learning activities and subjects. However in e-Learning 2.0 scenario, the number of learning activities and subjects tends to be very high, and thus we need to look for more flexible approaches. Metaheuristics have been used extensively in the literature obtaining satisfactory results for all kind of problems. Among metaheuristics, Memetic Algorithms have become one of the most used approaches due to their ability to combine a high exploration of the search space and a high exploitation of the most promising regions. In this paper we propose an adaptive memetic algorithm able to adapt to the different scenarios of LPP and obtain high-quality personalized learning experiences.

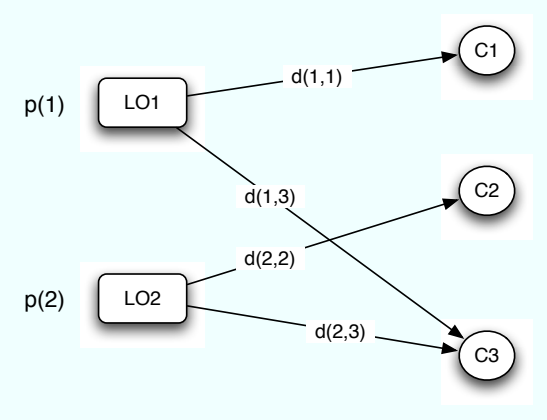

Fig. 2. Formalization of LPP.

IV. A Novel Memetic Approach to Solve Learning Presentation Problem

This section introduces a novel multi-agent memetic algorithm capable of efficiently solving the LPP problem and, consequently, supporting e-learning environments to define high-quality personalized learning experiences.

A. An Introduction to Memetic Algorithms

Optimization problems have focused the interest of the research community for a long time. For that reason several strategies have been developed in order to solve them in a reasonable computational time and find solutions with a near optimum quality. Among these strategies, *metaheuristics* play a fundamental role. Metaheuristics can be defined as, [33]: *"Intelligent strategies to design or improve very general, high performance heuristic procedures"*. Recent literature, e.g., [33],[37], reveals a wide variety of problems and methods that appear within the overall topic of heuristic optimization. Among these optimization methods, in the last years, Evolutionary Algorithms (EAs) have deeply stimulated research community.

EAs are an interdisciplinary research field with relationships to biology, artificial intelligence and numerical optimization and they are exploited in several engineering and scientific disciplines [34],[35],[36]. Nevertheless, although these algorithms have been exploited to try to resolve some of the more complex NP-complete problems, it is now well established that they are not well suited to fine tuning search in complex combinatorial spaces and, consequently, the hybridization with other techniques may greatly improve the efficiency of search [38]. Memetic Algorithms (MAs) are an extension of EAs that apply separate local optimization processes (hill climbing, simulated annealing, tabu search, etc.) to refine individuals. These hybrid methods [39] are inspired by models of adaptation in natural systems that combine the evolutionary adaptation of a population with individual learning of its members. The choice of name is inspired by Richard Dawkins' concept of a meme, which represents a unit of cultural evolution that can exhibit local refinement [40]. In the context of heuristic optimization, a meme is taken to represent a learning or development strategy. Thus, a memetic model of adaptation exhibits the plasticity of individuals that a strictly genetic model fails to capture.

From an optimization point of view [41], MAs have been shown to be both more efficient (i.e., requiring orders of magnitude fewer evaluations to find optima) and more effective (i.e., identifying higher quality solutions) than traditional EAs for some problem domains. As a result, MAs are gaining wide acceptance, in particular, in well-known combinatorial optimization problems where large instances have been solved to optimality and where other metaheuristics have failed to produce comparable results

However, despite the interesting results achieved by MAs, the process of designing effective and efficient MAs still raises some questions:

- What evolutionary strategies and local search methods can be applied within the memetic cycle?
- Which individuals in the population should be improved by local search, and how should they be chosen?
- How can the genetic operators best be integrated with local search in order to achieve a synergistic effect?

In fact these choices may greatly influence the search performance of MAs [42],[43],[44]. This evidence has led research community to develop MAs capable of adapting their behavior to the characteristics of the instance being solved, obtaining what has been given the name of third generation MAs or adaptive MAs.

From the different techniques included in adaptive MAs three approaches stand out on account of their results and popularity, namely hyperheuristics [45], co-evolution of memes [46] and meta-Lamarckian learning [47].

In this paper we propose an adaptive MA diverse from the previously explained, which uses fuzzy decision trees to model the knowledge automatically extracted from precedent executions of the metaheuristics in order to decide how to combine them to obtain better results.

B. A Novel Multi-agent Based Memetic Algorithm

In this section a Multi-Agent Memetic Algorithm (MAMA) is introduced. Agents and Multiagent systems are an interesting tool to model systems in which different entities interact. A software agent is a piece of software which represents an entity in a given environment in order to achieve certain

objectives in an autonomous and intelligent way, and that will possibly need to interact with other agents [48], [49]. One of the main aims of an agent is the intelligent resolution of complex problems, which are difficult to solve individually but can be solved by the interaction of different entities. This idea has lead to the definition of multiagent systems [50], [51], [48]. The purpose of these systems is to coordinate the different agents that compose them and to integrate individual objetives into a global objective. Their applications include many fields as: transportation, logistics, graphics, GIS, optimization problems...

The proposed multi-agent memetic algorithm (see fig. 3) is computed in two steps, dealing respectively with evolutionary and local methodologies. In the first step, a collection of different evolutionary optimization strategies (Genetic Algorithm, Particle Swarm Optimization, ...) are executed in a *parallel* way in order to locate the best region of problem's search space. Successively a set of local search strategies (Tabu Search, Simulated Annealing, ...) simultaneously exploits this region in order to find a high quality solution. In each step, the optimization strategies, *intelligently choose* their configuration parameters and *cooperate* by exchanging their solutions in *adaptive* way. The adaptability is achieved by means of a fuzzy rule base able to evaluate information extracted by a preliminary machine learning approach [52] that ranks computational performances related to evolutionary and local metaheuristics applied to a given optimization problem.

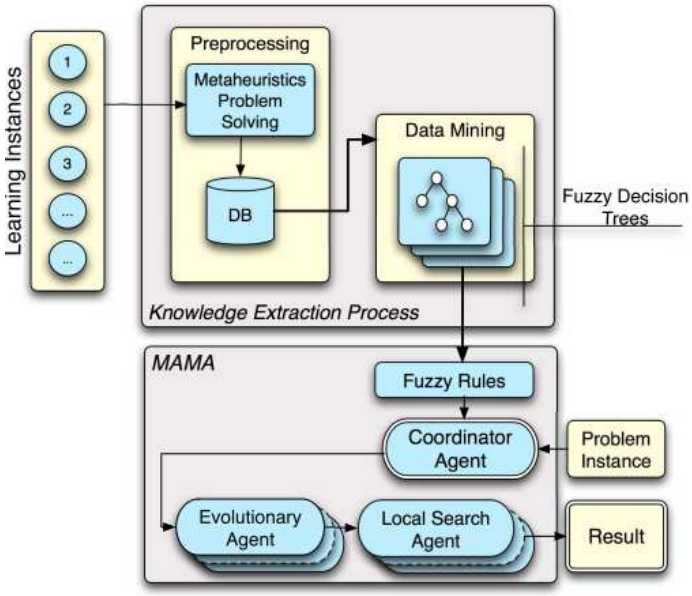

Fig. 3. Adaptive Memetic Architecture.

The parallel and distributed nature of our approach is fully suitable to be modeled using a multi-agent system where a set of so called *optimization agents* computes the cooperating metaheuristics under the supervision of a *coordinator agent* whose intelligence is provided by the aforementioned fuzzy rules and machine learning approach. The coordinator agent is responsible of initiating optimization process by parallel activating optimization agents and, in each phase, it coordinates the cooperation among optimization agents by choosing their configuration parameters and exchanging their solutions in an adaptive way.

```
void optimization_agent(n, q, m_i) begin
  1) if a message has been received from the
     coordinator replace poor solutions with the
     solutions contained in the message
  2) solution = compute metaheuristic m_i evaluating
     objective function n times;
  3) write solution on blackboard;
end
```

Fig. 4. Optimization Agent's Pseudocode

```
solution coordinator_agent(T, n, q, M) begin
  1) analyze the set of trees T in order to
     select the best parameter values for each
     metaheuristic;
  2) execute evolutionary phase:
       a) compute optimization_agent(n, q, m_i) in a
          parallel way for each optimization agent
          implementing an evolutionary strategy;
       b) exploit fuzzy rules, T and blackboard
          data to select a set of agents performing
          poorly;
       c) for each agent selected compose a message
          containing a collection of more suitable
          solutions and send it;
       d) repeat steps 2-a), 2-b), 2-c) until end
          condition is satisfied.
  3) bestsolution = read the best solution from
     blackboard;
  4) use bestsolution as initial solution for agents
     implementing local search;
  5) execute local search phase:
       a) compute optimization_agent(n, q, m_i) in a
          parallel way for each optimization agent
          implementing a local search strategy;
       b) exploit fuzzy rules, T and blackboard
          data to select a set of agents performing
          poorly;
       c) for each agent selected compose a message
          containing a collection of more suitable
          solutions and send it;
       d) repeat steps 5-a), 5-b), 5-c) until end
          condition is satisfied.
  6) return best solution computed;
end
```

Fig. 5. Coordinator Agent's Pseudocode

The adaptability task of coordinator agent is performed by considering knowledge coming from machine learning (see section IV-C) coded by means of a collection of fuzzy decision trees. In particular, trees' analysis supports coordinator agent to 1) choose the best metaheuristic parameters and 2) rank the metaheuristics suitability to solve the instance using numerical weights in the range $[0, 1]$. Through weights analysis, coordinator may realize that a metaheuristic is poorly performing and, consequently, it may update its solutions by adding a set of better solutions computed by other more suitable metaheuristics. This process allows poor metaheuristics to restart their optimization task in a more interesting point of the search space.

The cooperation is performed in the following way, optimization agents evaluate, during a certain period, the objective

function then they stop their computation in order to allow the coordinator agent to collect their information and make decisions. After that, each optimization agent continues exploring the search space but executing coordinator agent's orders.

To perform the communication between coordinator and optimization agents, a blackboard architecture will be used. Blackboards are data structures usually used as general communication mechanisms. In our proposal each agent has an assigned space on the blackboard where it periodically *writes* information about the solution it has found. The coordinator *reads* this information and directs the search of each agent.

Figures 4 and 5 show respectively the algorithms that control the behavior of optimization and coordinator agents by means of pseudocode.

1) Modeling intelligence of the coordinator agent: A crucial question of proposed memetic architecture is how to model the coordinator agent's intelligence in order to allow it to evaluate metaheuristics performances and control the cooperation among related optimization agents. This goal is achieved by using a set of fuzzy rules able to exploit the knowledge obtained by machine learning. More precisely, the coordinator agent uses fuzzy inference to:

- choose the best metaheuristics' parameters values;
- individuate when poor evolutionary metaheuristics have to receive a collection of individuals from other strategies that are showing a better behavior.
- individuate when local search methods have to reinitiate their search using the solution of another one which is showing a better behavior.

Coordinator agent implements first behavior by using a collection of fuzzy decision trees, as many as metaheuristics, obtained through machine learning. These trees analyze the instance in order to infer the most suitable parameters values for each metaheuristic. Moreover, the coordinator agent realizes the remaining two behaviors by considering two additional trees coming from machine learning. In detail, these trees analyze the instance and respectively rank evolutionary metaheuristics and local search metaheuristics, by returning a weights collection Ω where each $\omega_i \in \Omega$ takes a value in the range $[0, 1]$, the sum of the weights of evolutionary metaheuristics is 1, as the sum of the weights of local search metaheuristics. In short, each weight ω_i is associated with a metaheuristic and represents its suitability to solve the instance under consideration , i.e., the higher the weight of a metaheuristic the more suited it is.

In each phase, the coordinator agent uses a collection of fuzzy rules (one for each metaheuristic) exploiting Ω in order to derive the collection of poor strategies. For the sake of simplicity the rule shown below is related to evolutionary strategies (however rules acting in local phase are equivalent):

IF [($\omega_1 \cdot d_1$ OR ... OR $\omega_{|E|} \cdot d_{|E|}$) IS *enough*] THEN change the current solution of m_h.

where:

- m_h is the metaheuristic being evaluated by the rule;
- $d_i = (\xi(m_i) - \xi(m_h))/\max(\xi(m_i), \xi(m_h))$, where ξ is a measure of performance shown by metaheuristics. It is defined by the user;
- $\omega_i \in [0, 1]$ is the weight of m_i;
- $enough$ is a fuzzy set with trapezoidal membership function defined by a quadruplet (a, b, c, d) and whose universe of discourse is the range $[0, 1]$.

The main goal of each rule is to change the position in the search space of a metaheuristic which is showing a bad performance for a position near the solution of another metaheuristic with a better behavior. It should be noted that this rule tries to solve the problem that appears in parallel strategies where the exchange of solutions is unrestricted [53] and which tend to prematurely converge to local optimums characterized by a low quality.

To perform the firing of the fuzzy rules, $\alpha - cuts$ are used. In other words, only those rules whose activation value exceeds the defined $\alpha - cut$, are fired. In the event that more than one rule is activated then the coordinator applies all of them. In this way, the coordinator agent is capable of simultaneously adapt the behavior of several poor metaheuristics.

The metaheuristics' solution exchange is performed by taking into account two different situations: 1) a single antecedent clause of rule is fired or 2) many clauses are fired during inference process. In detail, let S be the collection of strategies related to fired clauses and let m_h be the poor metaheuristic that have to receive better solutions. If a single clause is fired then $|S| = 1$ otherwise $|S| > 1$.

Then, during the first phase (evolutionary phase), the poor metaheuristics' solutions changing is performed as follows:

1) $|S| = 1$. A proportion of the worst individuals of m_h is substituted by a set of solutions consisting of the best members of the population of strategy $m_k \in S$. The proportion is equals to ω_k.
2) $|S| > 1$. A proportion of the worst individuals of m_h is replaced by a set of solutions where each meta-heuristic chooses its solutions as said before. The proportion is equals to $\sum_{i=1}^{|S|} \omega_i$.

Differently from first phase, during the local optimization phase the changing is performed as follows:

- A solution "near" to the best solution obtained among the meta-heuristics $m_k \in S$ is sent to m_h. Where "near" means that the best solution is changed by applying, $\lceil \frac{1}{(2 \cdot \omega_k)} \rceil$ times, a mutation operator that depends on the problem. This operation is performed in order to introduce some diversification, since the exchange of identical solutions can cause that all metaheuristics perform the search in the same space.

C. MAMA Configuration through Machine Learning

As previously mentioned the adaptability of the proposed MA is provided by the knowledge obtained through an extraction process introduced in this section. The aim of the knowledge extraction process is to evaluate the performances of metaheuristics composing the MA when applied to instances of the problem under consideration. This process returns a set of trees containing: two trees respectively modeling the suitability of the evolutionary and local search strategies by providing the aforementioned weights Ω and, on the other

hand, one tree for each metaheuristic which provides the most suitable parameters choice for each metaheuristic.

The Knowledge extraction process is subdivided in two subphases, as can be seen in figure 3, Preprocessing and Data Mining.

1) Preprocessing: In preprocessing phase, metaheuristics that compose the system are applied to a set of training instances in order to build a collection of databases, named refined databases, that will be exploited by data mining to compute the trees that guide the coordinator. The obtaining of these databases is carried out in two steps. First, information about the performance of metaheuristics is stored in a set of so called *raw databases*, where each raw database is associated to a different metaheuristic. Successively, data contained in raw databases are processed in order to compute the final refined databases, which contain additional information useful to extract significant knowledge during data mining.

In particular, the obtaining of raw databases is carried out in the following way. For each metaheuristic and each parameter configuring this metaheuristic, a set of parameter values is chosen, after that each metaheuristic is executed k times with each possible combination of parameter values, being k an *iterative factor*, i.e., a predefined value used to obtain more accurate performance estimations. For each one of these executions a row is inserted into the raw database related to the metaheuristic being evaluated. In details, a raw database row contains:

- a description of the specific instance of the problem;
- the values of the parameters used in the execution of each metaheuristic;
- the final solution obtained.

Next step is to analyze raw databases in order to compute the refined databases. Refined databases are smaller than raw databases and contain additional information useful to extract significant knowledge during data mining. In detail, for each metaheuristic a refined database is computed which contains information about its best parameters combinations, i.e., which parameters values obtained a better performance. Besides, two more refined databases are computed which respectively contain information about the weights Ω associated to evolutionary and local search strategies.

In order to build databases concerning parameters for each metaheuristic its raw database is processed as follows:

1) for each instance and for each parameters combination calculate the average fitness value of its k executions.
2) for each instance select the parameter combination computing the best average fitness value;
3) the refined database is updated with a new entry containing the description of instance and the best parameter combination.

The remaining two databases are directly related to evolutionary strategies and local search methods. In detail, the refined database related to evolutionary methods is processed as follows:

1) for each metaheuristic, for each instance and for each parameters combination calculate the average fitness value of its k executions.
2) for each metaheuristic and each instance, select the best average fitness value and compute the *metaheuristic suitability weight* ω_i as:

$$\omega_i = \frac{fit^i_{best}}{\sum_{l=1}^{\#evolutionary\_strategies} fit^l_{best}}$$

3) for each metaheuristic and each instance the refined database is updated with a new entry containing the description of the instance and the metaheuristic suitability weight ω_i.

Database associated to local search methods is analogously built by replacing evolutionary strategies with local search methods.

Refined databases represent the output result of Preprocessing phase and they will be used by a data mining technique to complete the Knowledge Extraction Process in computational efficient way.

2) Data Mining: Once gathered performance information and collected it through refined databases, the Data Mining phase extracts the collection of knowledge models through data mining. Our data mining approach exploits a fuzzy decision tree (FDT) technique [54] to extract knowledge from refined databases and build the aforementioned collection of trees. Two main reasons have guided the choice of FDTs. First their interpretability, decision trees stand out on account of their simplicity and readability, which is an important issue in data mining, and enable a full understanding of the knowledge that guides the coordinator. Second, their fuzziness, i.e., the possibility of dealing with fuzzy theory, which can improve inference performance by means of approximate reasoning and provide an easy way to obtain the weights Ω, one of the most important parts of coordinator's knowledge.

A summary of Data Mining phase is shown in Fig. 6, which conceptually illustrates the obtaining of the fuzzy decision trees and their structure. Once this phase is finished the collection of fuzzy decision trees that models coordinators's knowledge, is obtained, and it is ready to control the execution of the optimization agents.

V. Experimental Results

In this section we want to asses the validity of the proposed approach, MAMA. In order to do that we will perform some preliminary tests solving different instances artificially generated. However, the main purpose of MAMA is to help improving the e-learning process. For this reason we plan to include it as a plug-in of a learning management system. Consequently, before obtaining the final plug-in, we will perform a test with real learners, which will indicate the real impact of MAMA in the process of e-learning.

A. A Plug-in to Personalize e-Learning Experiences

As mentioned above, the main purpose of MAMA is to help improving the e-learning experiences, and for this reason we consider interesting its inclusion on a learning management system. In particular, we plan to include it in Sakai project[2].

[2]www.sakaiproject.org

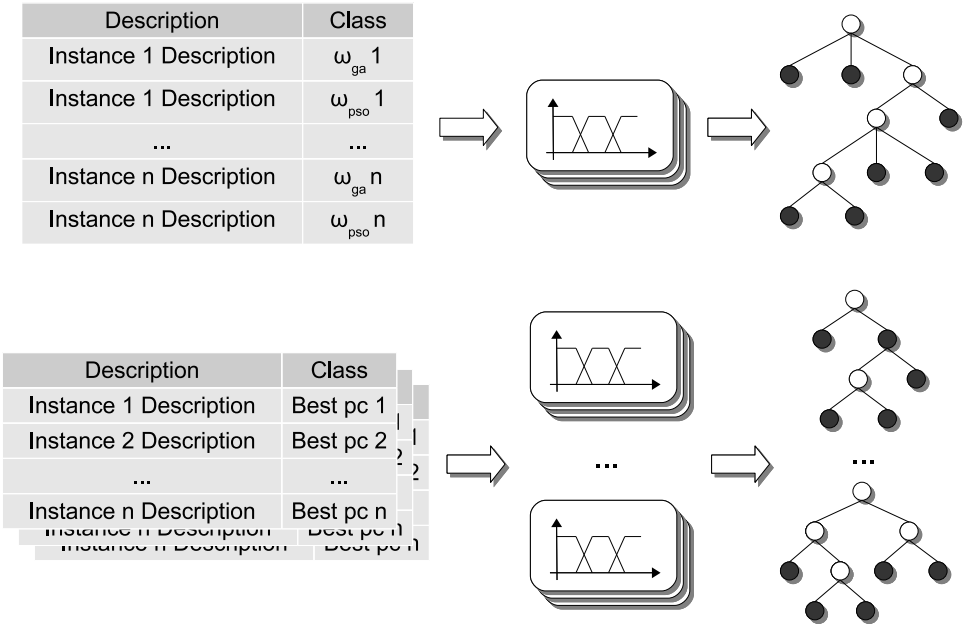

Fig. 6. Data Mining Phase.

Sakai is an open source learning management system developed by the world's leading educational institutions and used by over 150 institutions. It enables powerful teaching, learning and research collaboration within a feature-rich, enterprise platform. However, Sakai does not provide the possibility of automatically personalize e-learning experiences in the previously explained way. We consider this an essential feature for modern e-learning solutions, and thus believe that the development of a plug-in that adds this feature can improve learners' experience.

B. Configuring MAMA to solve LPP

1) Metaheuristics Included in the Plug-in: A crucial issue in the design of our plug-in is the choice of the metaheuristics that will made up the system. There is a host of different metaheuristics which can obtain good results, but for the problem being considered we have chosen four strategies, two evolutionary metaheuristics, a Genetic Algorithm (GA) [55] and a Particle Swarm Optimization (PSO) [56], and two local search metaheuristics, a Tabu Search (TS) [57] and a Simulated Annealing (SA) [58]. Other metaheuristics could have been chosen or even added, but as a first step we have chosen these four.

Concerning the implementation of the metaheuristics, everyone was programmed using Java, being the evolutionary strategies obtained from CIlib [59],[60]. With regard to the fuzzy engine used to model the coordinator we utilized Fuzzy Markup Language [61]. With this tool we modeled the different rules and obtained a fuzzy engine that finally was slightly modified to obtain an engine appropriate to our purpose.

2) Applying the Knowledge Extraction Process phase: The first step in the construction of Sakai plug-in is the acquisition of the knowledge that will guide the coordinator. As it was previously mentioned in order to acquire this knowledge we have to apply a knowledge extraction process divided in two phases, Preprocessing and Data Mining.

In the first phase, the databases from which knowledge shall be extracted are collected. The application of this phase required the use of several instances of above said e-learning LPP problem. The resolution of these instances using the previously chosen metaheuristics with different combinations of parameters values provided the databases from which the knowledge shall be extracted. Specifically, each LPP instance was solved 5 times with each metaheuristic and each combination of parameters.

In particular each metaheuristic used the following parameter values:

- Genetic Algorithm:
  - Selection operator: Roulette wheel and Tournament.
  - Cross probability: 0.6 and 0.4.
  - Mutation probability: 0.01 and 0.001
  - Population size: 100.
- Tabu Search:
  - Tabu list size: 10, 100 and 1000.
  - Diversification threshold: 1, 10 and 50.
- Particle Swarm Optimization:
  - Number of particles: 100.
  - Topology: Local best considering 4 neighbors, Local best considering 2 neighbors, global best and Von Neumann.
  - ω: 0,729844.
  - r_p: 1.49 and 2.
  - r_g: 1.49 and 2.
- Simulated Annealing:
  - Cooling schedule: Hyperbolic tangent, hyperbolic cosine, square root and exponential.
  - Inner iterations: 5000, 7000, 9000 and 10000.

TABLE II
PARAMETERS VALUES FOR EACH METAHEURISTIC

GA		*PSO*	
parameter	*value*	*parameter*	*value*
Selection op.	Roulette	Topology	Local Best 2N
Cross prob.	0.6	ω	0.729844
Mutation prob.	0.01	r_p	2
Population size	100	r_g	2
		# of particles	100

TS		*SA*	
parameter	*value*	*parameter*	*value*
Tabu list size	10	Cooling schedule	Hyperbolic Tangent
Diversification t.	50	Inner iterations	5000

After the obtaining of these databases, we are prepared to apply Data Mining phase and obtain the fuzzy decision trees that configure the coordinator. In order to do that we applied the learning algorithm to these databases obtaining two trees that will provide the weights for the different metaheuristics, and one tree for each metaheuristic that will provide the parameters to be used.

3) Integrating knowledge in coordinator's model: After generating the fuzzy decision trees we have to integrate them with the fuzzy set of rules in order to produce the intelligence of the coordinator. Certain decisions need to be made concerning some parameters of the set of fuzzy rules, namely: define the measure of performance used by the rules, the fuzzy set $enough$ and choose the $\alpha - cut$ better suited for the problem. The following options have been chosen:

- The performance measure, ξ in rule definition, will be the value of the objective function.
- The fuzzy set $enough$ will be have a trapezoidal membership function where $(a, b, c, d) = (0, 0.01, 1, 1)$.
- The $\alpha - cut$ will be set to 0.75.

C. Computational experiments

Hereafter, we perform some tests in order to assess the validity of MAMA, comparing its results with those obtained by the different non-adaptive MAs that can be obtained from the combination of the single metaheuristics that compose MAMA in a sequential way, namely:

- Genetic Algorithm + Tabu Search (GA+TS).
- Genetic Algorithm + Simulated Annealing (GA+SA).
- Particle Swarm Optimization + Tabu Search (PSO+TS).
- Particle Swarm Optimization + Simulated Annealing (PSO+SA).

The parameters used by each metaheuristic are shown in Table II.

Regarding execution, for the cooperative strategies one can resort to parallel schemes if time is important, or one can simulate the parallelism in a one-processor computer. This is the strategy taken here and the procedure is extremely simple. We construct an array of solvers and we run them using a round-robin schema. This implementation uses an asynchronous communication mode that is simulated in this way: solvers are executed during a certain number of evaluations of the objective function. This amount of evaluations is a random number in a given interval and after this period of time, information exchanges are performed, in particular for these experiments we will use $[100, 150]$. These steps are repeated until the stop condition, given in terms of objective function evaluations, is fulfilled.

In terms of time, the execution of coordinator agent only requires the firing of the set of rules, and with the frequency currently used, the time used by coordinator agent is negligible.

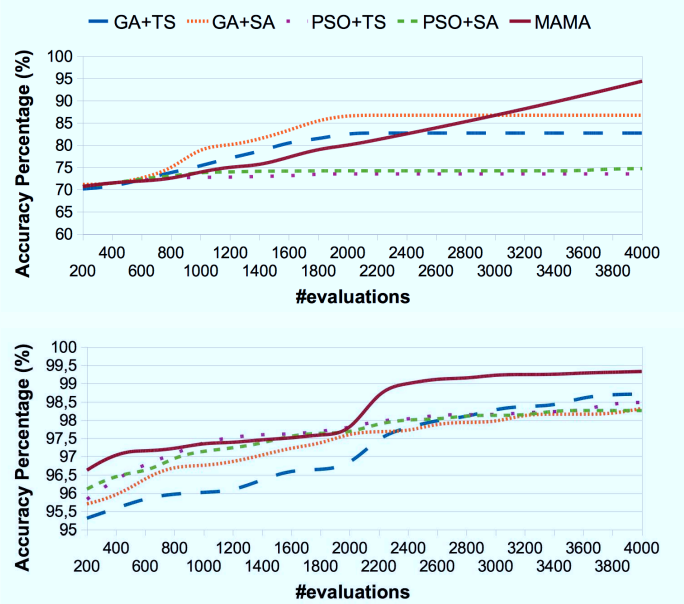

Fig. 7. Evolutions of the different strategies.

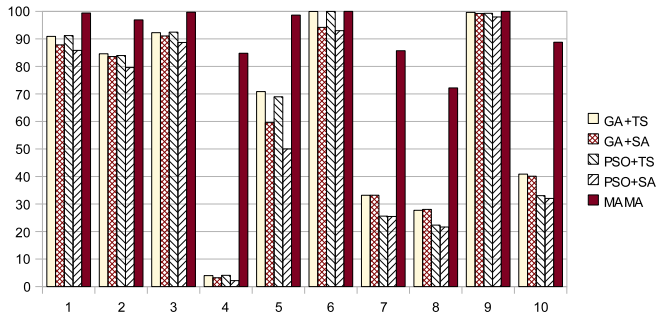

Fig. 8. Performance of the different strategies.

1) Performance tests: As a first approach it would be interesting to perform some preliminary tests in order to asses the validity of the proposed strategy. In order to perform these tests we have created an instance generator able to construct artificial instances of the learning presentation problem. We have identified some instance's features that may influence the behavior and performance of different solving strategies:

- Number of learners.
- Ratio between learning objects and learners.

- Connection percentage between learning objects and learners.

Initially we have carried out two tests with two different instances generated using the aforementioned generator. The purpose of these tests was to graphically interpret the behavior of the different strategies during their search of the optimum. The results are shown in Fig. 7, this figure illustrates the evolution of the 5 strategies. The graphics plot the fitness value obtained by each strategy at each moment of the execution, showing the average of 10 executions of 4000 evaluations. From these graphics we can extract interesting information about the behavior of MAMA. In particular, it can be observed that although MAMA's convergence is not the fastest, it obtains in both instances the best performance, these are promising results which seem to indicate that MAMA solves e-learning LPP problem in a more efficient way than other static memetic approaches and, as will be shown afterwards, it generates very suitable personalized learning experiences.

After these first executions we have performed more extensive tests in order to carry out a statistical analysis of the results. As a test bed we used 10 different instances obtained using the previously defined instance generator. Each one of them was solved 10 times using 40000 evaluations of the objective function, note that the number of evaluations is incremented with regard to the previous tests, because the former are just preliminary tests and in the new tests we want more stable results. Final results are shown in Fig. 8. At first glance, MAMA clearly outperforms non adaptive MAs. However, in order to check this first intuition we will perform an analysis of methods using statistical techniques.

Following the methodology of [62], which uses non-parametric tests, we use the Wilcoxon signed-rank test to compare two methods. This test is a non-parametric statistical procedure for performing pairwise comparisons between two algorithms. This is the analogous of the paired t-test in non-parametric statistical procedures; therefore, it is a pairwise test that aims to detect significant differences between two sample means, that is, the behavior of two algorithms. When we compare multiple methods, we use the Friedman test and the Benjamini-Hochberg method as post-hoc test. This last method is more powerful than Bonferroni-Dunn test, Holm test and Hochberg method. Friedman test is a non-parametric test equivalent to the repeated-measures ANOVA. Under the null-hypothesis, it states that the algorithms are equivalent, so a rejection of this hypothesis implies the existence of differences among the performance of all the algorithms studied. After this, a post-hoc test (we use the Benjamini-Hochberg method) could be used in order to find whether the control or proposed algorithm presents statistical differences with regards to the remaining methods in the comparison.

In this case, Friedman test indicates that algorithms are not equivalent with a significance level of the 99.9%. Once we have test this point we check the differences between MAMA and the non-adaptive MAs. After applying Wilcoxon test and adjusting it using Benjamini-Hochberg method we observe that the results obtained by MAMA are better than the rest of results with a significance level of the 99%.

TABLE III
RESULTS FOR REAL LEARNERS.

	Without MAMA			
	before e-learning		after e-learning	
learner	quantification	level	quantification	level
1	8	High	8	High
2	9	High	9	High
3	4	Medium	6	Medium
4	3	Low	5	Medium
5	5	Medium	5	Medium
6	6	Medium	8	High
7	5	Medium	6	Medium
8	6	Medium	8	High
9	5	Medium	6	Medium
10	5	Medium	7	Medium
11	5	Medium	6	Medium
12	3	Low	4	Medium
13	1	Low	4	Medium
14	1	Low	6	Medium
15	0	Low	2	Low
16	2	Low	3	Low
17	4	Medium	6	Medium
18	1	Low	2	Low
19	7	Medium	8	High
20	1	Low	2	Low
	With MAMA			
21	6	Medium	8	High
22	2	Low	6	Medium
23	6	Medium	9	High
24	4	Medium	7	Medium
25	8	High	9	High
26	2	Low	3	Low
27	2	Low	7	Medium
28	4	Medium	6	Medium

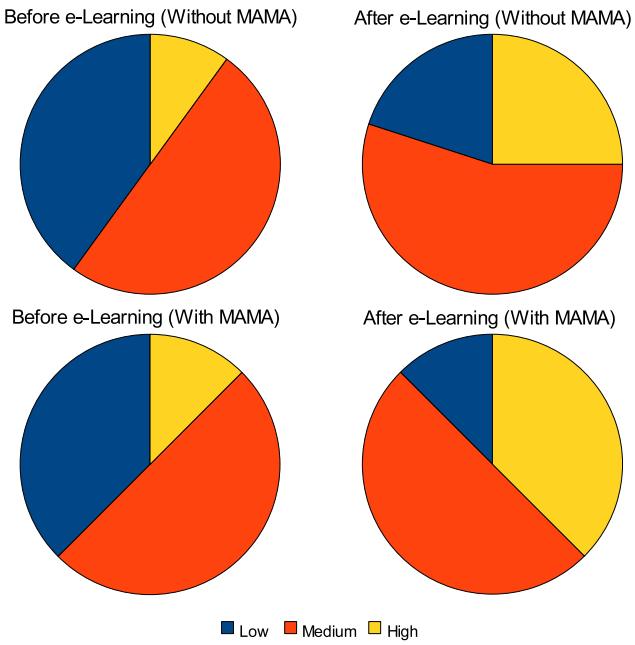

Fig. 9. Experimentation results on a group of 28 learners with and without the proposed memetic Sakai plug-in prototype.

2) Learning Comparative Studies: Once we have demonstrated the effectiveness of MAMA with artificial instances, we perform a real world test, involving real learners that want to

train on enterprise management. The control and experimental groups were made up of 20 and 8 learners respectively. Data corresponding to the control group comes from historical data from a previous experiment, and shows the results of learning using Sakai without the help of MAMA. On the other hand, data corresponding to experimental group was obtained from the results achieved by a new group of learners, which were able to use the capabilities of MAMA. All the voluntary learners were tested before and after a training phase on the same topics. In all the tests the learners' skills in the chosen domain were quantified using three ability ranges: low-level (0-3 scores), medium-level (4-7 scores) and high-level (8-10 scores). Table III presents the results obtained by each learner involved in the experiment, and Figure 9 presents a summary of the results. As can be observed, both e-Learning approaches provide the necessary tools to improve learners' knowledge, in fact the quantifications of the knowledge are statistically better with a significance of the 99.9% after using e-learning. Moreover those learners that have used MAMA have obtained better results than those which not, with an statistical significance of the 95%. These results are very promising and support the creation of a plug-in for Sakai able to include MAMA's capabilities.

VI. Conclusions

Recent advances in Web 2.0 and its tools (Blogs, Wikis, Podcast, Web Sharing Applications etc...) have impulsed the development of the so called e-learning 2.0. In this context, Learning Management Systems should promote new e-learning trends by means of suitable models and processes that dynamically and intelligently create personalized e-learning experiences adapted to learner expectations and objectives. This result can be achieved by defining personalized e-learning experiences, intended as most fitting sequences of learning activities able to maximize the understanding level of learners with respect to specific learning objectives, a representation model to describe both educational domains and learners characteristics is required. In this scenario, ontological methodologies model and represent the e- learning contexts realizing a conceptualization of an educational domain by identifying its relevant subjects and organizing them by means of a fixed set of relations. Analysis of ontological relationships enables a convenient way to bind subjects with learning activities in order to define personalized e-learning experiences. Our work proposed an innovative memetic algorithm capable of efficiently generating learning presentations. In detail, we have proposed a parallel cooperative adaptive memetic algorithm aimed to personalize e-learning environment by combining and configuring cooperative evolutionary and local search metaheuristics in order to improve overall performance.

The cooperation among the different optimization strategies is performed by intelligently exchanging solutions in precise moments and under certain conditions. The exchange control is supervised by a set of fuzzy rules, which exploits knowledge modeled by a collection of fuzzy decision trees and obtained using a preliminary learning process. With this knowledge the fuzzy engine can predict the behavior of each metaheuristic and adapt it to the presentation being personalized. The parallel and distributed nature of our approach is fully suitable to be modeled using a multi-agent system where software agents compute the cooperating metaheuristics under the supervision of a coordinator agent whose intelligence is given from the aforementioned fuzzy rules and decision trees.

The aim of the proposed multi-agent system is to improve the capabilities of Learning Management Systems providing an integrated approach towards achieving the personalization in e-learning environments. For that reason, we have designed a plug-in for an important open-source Learning Management System, Sakai. Our proposal enhances significantly the learning framework in terms of flexibility and efficiency. In particular, it can support various personalization levels (intended as most fitting sequences of learning activities able to maximize the understanding level of learners with respect to specific learning objectives) exploiting machine-understandable representations of educational domains and learners characteristics.

Acknowledgements

The authors thank the MICINN of Spain and the "Fondo Europeo de Desarrollo Regional" (FEDER) for their support given to this work under the project TIN2008-06872-C04-03. Thanks also to the Funding Program for Research Groups of Excellence with code 04552/GERM/06 by the "Fundación Séneca". E. Muñoz is supported by the scholarship program FPI from the "Fundación Séneca".

References

[1] A. Romiszowski, "Hows the E-learning Baby? Factors Leading to Success or Failure of an Educational Technology Innovation," *Educational Technology* vol. 44, pp. 5–27, 2004.

[2] M. Ebner, "E-Learning 2.0 = e-Learning 1.0 + Web 2.0?," *Availability, Reliability and Security, 2007. ARES 2007. The Second International Conference on*, pp. 1235–1239, 2007.

[3] D. Dagger, A. O' Connor, S. Lawless, E. Walsh, V.P. Wade, "Service-Oriented E-Learning Platforms: From Monolithic Systems to Flexible Services," *IEEE Internet Computing* vol. 11, pp. 28–35, 2007.

[4] P. Jen-Hwa Hu, W. Hui, T.H.K. Clark, and Kar Yan Tam, "Technology-Assisted Learning and Learning Style: A Longitudinal Field Experiment," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol.37, no.6, pp.1099-1112, Nov. 2007.

[5] T. Gruber, "Toward Principles for the Design of Ontologies Used for Knowledge Sharing," *International Journal of Human-Computer Studies* vol. 43, pp. 907–928, 1994.

[6] R. Valencia-Garcia, F. Garcia-Sanchez, D. Castellanos-Nieves, J.T. Fernández-Breis, "OWLPath: An OWL Ontology-Guided Query Editor," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol.41, no.1, pp.121-136, Jan. 2011.

[7] D. Dicheva, C. Dichev, "TM4L: Creating and Browsing Educational Topic Maps," *British Journal of Educational Technology*, vol. 37, pp. 391–404, 2006.

[8] G. Cornuejols, G. L. Nemhauser, L. A. Wolsey, "The uncapacited facility location problem," *Discrete Location Theory*, P. Mirchandani and R. Francis (Eds.), John Wiley and Sons Inc., New York, pp. 119–171, 1990.

[9] S. Murugesan, "Understanding Web 2.0," *IT Professional* vol. 9, pp. 34–41, 2007.

[10] A. Sutcliffe, A, C. Wei-Chung and R.S. Neville, "Applying Evolutionary Computing to Complex Systems Design," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol.37, no.5, pp.770-779, Sept. 2007.

[11] L. Aroyo, S. Pokraev, R. Brussee, "Preparing SCORM for the Semantic Web," *Proceedings of International Conference on Ontologies, Databases, and Applications of Semantics*, pp. 621–628, 2003.

[12] C. Qu, W. Neidl, "A Collaborative Courseware Generating System based on WebDAV, XML, and JSP," *IEEE Int. Conference on Advanced Learning Technologies*, pp. 197–198, 2001.
[13] W. Chen, W., Y. Hayashi, L. Jin, I. Mitsuru, R. Mizoguchi, "An Ontology-based Intelligent Authoring Tool," *Sixth International Conference on Computers in Education*, pp. 41–49, 1998.
[14] B. Simon, "Do e-learning standards meet their challenges?," *Proceedings of Workshops Standardisierung im eLearning*, pp. 1–15, 2002.
[15] R. Mizogushi, Y. Kitamura, "Ontology-based systematization of functional knowledge," *Journal of Engineering Design* vol. 15, pp. 327–351, 2004.
[16] V. Devedic, J. Jovanovic, D. Gaevic, "The Pragmatics of Current E-Learning Standards," *IEEE Internet Computing*, vol 3, pp. 19–27, 2007.
[17] A. Heinze, C. Procter, "Reections On The Use Of Blended Learning," *Proceedings of Education in a Changing Environment*, pp. 1–20, 2004.
[18] A.Chiu, C. C. Sheng, A. P. Chen, "Designing a Dynamic E-learning Project Performance Evaluation Framework," *Proceedings of 7th IEEE International Conference on Advanced Learning Technologies*, pp. 381–385, 2007.
[19] A. El Alami, N. Casel, and D. Zampunieris, "An Architecture for e-Learning System with Computational Intelligence," *Knowledge-Based Intelligent Information and Engineering Systems (LNCS)*, vol. 4693, pp. 58–65, 2007.
[20] C. E. Alexakos, K. C. Giotopoulos, E.J. Thermogianni, G. N. Beligiannis, S. D. Likothanassis, "Integrating E-learning Environments with Computational Intelligence Assessment Agents," *Proceedings of World Academy of Science, Engineering and Technology* vol. 13, pp. 233–238, 2006.
[21] S. Buraga, "Developing Agent-Oriented E-Learning Systems," *Proceedings of The 14th International Conference on Control Systems and Computer Science*, vol. 2, pp. 74–82, 2003.
[22] A. Angehrn, T. Nabeth, L. Razmerita, C. Roda, "K-inca: Using Artificial Agents for Helping People to Learn New Behaviors," *Proceedings of the IEEE International Conference on Advanced Learning Technologies*, pp. 225–226, 2001.
[23] A. Andronico, A. Carbonaro, G. Casadei, L. Colazzo, A. Molinari, M. Ronchetti, "Integrating a multi-agent recommendation system into a Mobile Learning Management System," *Proceedings of Artificial Intelligence in Mobile System*, pp. 123–132, 2003.
[24] D.A. Pelta, R.R Yager, "Decision Strategies in Mediated Multiagent Negotiations: An Optimization Approach," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol.40, no.3, pp.635-640, May 2010.
[25] O.R. Zaiane, "Building a Recommender Agent for e-Learning Systems," *Proceedings of the International Conference on Computers in Education*, pp. 55–59, 2002.
[26] V. Pankratius, O. Sandel, W. Stucky, "Retrieving Content with Agents in Web Service E-Learning Systems," *Proceedings of the First IFIP Conference on Artificial Intelligence Applications and Innovations*, pp. 341–348, 2004.
[27] G. Albano, M. Gaeta, S. Salerno, "E-learning: a model and process proposal," *International Journal of Knowledge and Learning*, vol.2, pp. 73–88, 2006.
[28] R. Koper, "Current Research in Learning Design," *Educational Technology & Society*, vol. 9, pp.13–22, 2006.
[29] N. Guarino, *Formal Ontology in Information Systems*, IOS Press, 1998.
[30] T. J. Kopcha, H. Sullivan, "Learner preferences and prior knowledge in learner-controlled computer-based instruction," *Educational Technology Research and Development*, vol. 3, pp. 265–286, 2008.
[31] G. Albano, M. Gaeta, P. Ritrovato, "Testing hypotheses in the functional linear model," *Scandinavian Journal of Statistics*, vol. 30, pp. 241–251, 2007.
[32] G. Acampora, M. Gaeta, V. Loia, P. Ritrovato, S.Salerno, "Optimizing Learning Path Selection through Memetic Algorithms," *IEEE International Joint Conference of Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence)*, pp. 3869–3875, 2008.
[33] B. Melián, J.A. Moreno, J.M. Moreno, "Metaheursticas: una visión global," *Revista Iberoamericana de Inteligencia Artificial*, vol. 19 pp. 7–28, 2003.
[34] Huai-Kuang Tsai, Jinn-Moon Yang, Yuan-Fang Tsai, Cheng-Yan Kao, "An evolutionary algorithm for large traveling salesman problems," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* , vol.34, no.4, pp.1718-1729, Aug. 2004
[35] Y. Xiong, B. Golden, E. Wasil, "A one-parameter genetic algorithm for the minimum labeling spanning tree problem," *Evolutionary Computation, IEEE Transactions on* , vol.9, no.1, pp. 55-60, Feb. 2005
[36] S. Yussof, H. S. Ong, "Finding multi-constrained path using genetic algorithm," *in* Proc. Telecommunications and Malaysia International Conference on Communications, 2007. ICT-MICC 2007. IEEE International Conference on, pp.713-718, May 2007.
[37] F. Glover, G.A. Kochenberger, *Handbook of Metaheuristics*, Kluwer Academic, Dordrecht, 2003.
[38] D. Wolpert and W. Macready, "No free lunch theorems for optimization," *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82 1997.
[39] M.M.O. Sidi, S. Hammadi, S. Hayat and P. Borne, "Urban Transport Network Regulation and Evaluation: A Fuzzy Evolutionary Approach," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol.38, no.2, pp.309-318, March 2008.
[40] R. Dawkins, *The Selfish Gene*. New York: Oxford Univ. Press, 1976.
[41] N. Krasnogor, J.E. Smith, "A tutorial for competent memetic algorithms: Model, taxonomy and design issues." *IEEE Trans. Evol. Algorithms* vol. 9 no. 5, pp. 474–488, 2005.
[42] E. K. Burke, G. Kendall, and E. Soubeiga, "A tabu search hyperheuristic for timetabling and rostering," *J. Heuristics*, vol. 9, no. 6, pp. 451–470, 2003.
[43] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Trans. Evol. Comput.*, vol. 7, pp. 204–223, 2003.
[44] N. Zhu, Y. S. Ong, K. W. Wong, and K. T. Seow, "Using memetic algorithms for fuzzy modeling," *Austral. J. Intell. Inform. Process.* vol. 8, no. 3, pp. 147–154, 2004.
[45] N. Krasnogor, B. Blackburne, J. D. Hirst, and E. K. N. Burke, "Multimeme algorithms for the structure prediction and structure comparison of proteins," *Parallel Problem Solving From Nature,Lecture Notes in Computer Science*, pp. 769–778, 2002.
[46] J. E. Smith et al., "Co-evolving memetic algorithms: Initial investigations," *Parallel Problem Solving From Nature PPSN VII, G. Guervos et al., Eds.* Berlin, Germany: Springer, vol. 2439, Lecture Notes in Computer Science, pp. 537–548, 2002.
[47] Y. S. Ong, A. J. Keane, "Meta-Lamarckian in memetic algorithm," *IEEE Trans. Evol. Comput.*, vol. 8, pp. 99–110, 2004.
[48] M.J. Wooldridge, *Multi-agent Systems : An Introduction*, John Wiley and Sons, 2002.
[49] M. Luck, P. McBurney, and C. Preist, *Agent technology: Enabling next generation computing: a roadmap for agent based computing*, Agentlink, 2003.
[50] N.R. Jennings, K.P. Sycara, and M. Wooldridge, *A roadmap of agent research and development*, Autonomous Agents and Multi-Agent Systems **1** (1998), no. 1, 7–38.
[51] G. Weiss (ed.), *Multiagent systems: a modern approach to distributed artificial intelligence*, MIT Press, 1999.
[52] J.M. Cadenas, M.C. Garrido, E. Muñoz, "Using machine learning in a cooperative hybrid parallel strategy of metaheuristics," *Information Sciences*, vol.179, pp. 3255–3267. 2009.
[53] T. G. Crainic, M. Gendreau, P. Hansen, N. Mladenovic, "Cooperative parallel variable neighborhood search for the p-median," Journal of Heuristics, vol.10, pp. 293–314, 2004
[54] C.Z. Janikow, "Fuzzy decision trees: issues and methods," IEEE Transaction System, Man, and Cybernetics, Part B, vol. 28, no. 1, pp. 1–14, 1998
[55] C. Reeves, *Genetic Algorithms*, Handbook of Metaheuristics, Kluwer Academic, 2003.
[56] R.C. Eberhart, Y. Shi, and J. Kennedy, *Swarm intelligence (the morgan kaufmann series in artificial intelligence)*, 1st ed., Morgan Kaufmann, April 2001.
[57] M. Gendreau, *An Introduction to Tabu Search*, Handbook of Metaheuristics, Kluwer Academic, 2003.
[58] D. Henderson, S.H. Jacobson, and A.W. Johnson, *The theory and practice of simulated annealing*, Handbook of Metaheuristics, Kluwer Academic, 2003.
[59] G. Pampará, A.P. Engelbreght, T. Cloete, "CIlib: A Collaborative Framework for Computational Intelligence Algorithms - Part I," in Proceeding for 2008 International Joint Conference on Neural Networks (IJCNN 2008), Hong Kong (2008) pp. 1751–1758.
[60] G. Pampará, A.P. Engelbreght, T. Cloete, "CIlib: A Collaborative Framework for Computational Intelligence Algorithms - Part II," in Proceeding for 2008 International Joint Conference on Neural Networks (IJCNN 2008), Hong Kong (2008) pp. 1765–1774
[61] G. Acampora, V. Loia, "Fuzzy control interoperability and scalability for adaptive domotic framework," *Industrial Informatics, IEEE Transactions on*, Vol. 1, No. 2, pp. 97 – 111, 2005.

[62] S. García, A. Fernández, J. Luengo, F. Herrera, A study statistical of techniques and performance measures for genetics-based machine learning: accuracy and interpretability, *Soft Computing* 13 (10), 959–977 (2009).