A review of message passing algorithms in estimation of distribution algorithms

**Roberto Santana · Alexander Mendiburu ·
Jose A. Lozano**

Abstract Message passing algorithms (MPAs) have been traditionally used as an inference method in probabilistic graphical models. Some MPA variants have recently been introduced in the field of estimation of distribution algorithms (EDAs) as a way to improve the efficiency of these algorithms. Multiple developments on MPAs point to an increasing potential of these methods for their application as part of hybrid EDAs. In this paper we review recent work on EDAs that apply MPAs and propose ways to further extend the useful synergies between MPAs and EDAs. Furthermore, we analyze some of the implications that MPA developments can have in their future application to EDAs and other evolutionary algorithms.

Keywords Message passing algorithms · Propagation · Estimation of distribution algorithms · Hybridization

1 Introduction

One of the reasons that have been used to explain the success of genetic algorithms (GAs; Goldberg 1989; Holland 1975) is the systematic creation and parallel recombination of partial solutions or building blocks. From that point of view GAs can be seen as automatic problem decomposition procedures in which information about promising partial solutions is synthesized in multiple ways and refined along generations. However, it has been reported that in many difficult problems blind crossover recombination is not an appropriate choice since partial good solutions tend to be disrupted. Heuristic crossover can alleviate these situations but usually human expertise and a priori information about the problem characteristics are required.

Estimation of distribution algorithms (EDAs; Larrañaga and Lozano 2002; Lozano et al. 2006; Mühlenbein and Paaß 1996; Pelikan et al. 2002) are evolutionary algorithms (EAs) that learn a probabilistic model from the selected solutions and use this model to sample new solutions. They have been praised for their capacity to capture and exploit the interactions between the problem variables, limiting to a large extent the disruption of partial solutions. EDAs can also be seen as a more comprehensive approach to automatic problem decomposition in EAs than in GAs. The reason is that they are able to explicitly represent the problem structure which manages to produce in each generation a candidate model for problem decomposition.

The introduction of probabilistic models that explicitly describe the interactions between the components of the problem has led to the conception of more efficient EAs. However, the way in which problem decomposition is exploited to generate new solutions remains very similar in GAs and EDAs. Both algorithms implement the mixing of promising partial solutions during the generation step. Mixing is supposed to be more efficient in EDAs because the structure of the probabilistic model allows to organize the generation of the solutions in such a way that the interactions are respected.

R. Santana (✉) · A. Mendiburu · J. A. Lozano
Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country (UPV/EHU), Paseo Manuel de Lardizabal 1, 20080 San Sebastián, Guipúzcoa, Spain
e-mail: roberto.santana@ehu.es

A. Mendiburu
e-mail: alexander.mendiburu@ehu.es

J. A. Lozano
e-mail: ja.lozano@ehu.es

Algorithm 1 Estimation of
distribution algorithm

1	Set $t \Leftarrow 0$. Generate M solutions randomly.
2	**do** {
3	Evaluate the solutions using the fitness function.
4	Select a set D_t^S of $N \leq M$ solutions according to a selection method.
5	Compute a probabilistic model from D_t^S.
6	Generate M new solutions sampling from the distribution represented in the model.
7	$t \Leftarrow t + 1$
8	} **until** Termination criteria are met.

EDAs also have an associated computational overhead which is produced by the procedures needed to learn the probabilistic models from the data and generate new solutions sampling from them. Finding a fine balance between the computational cost incurred by the use of a particular probabilistic model and its contribution to the search efficiency is a key issue for the successful application of EDAs. Since the influence of EDAs' computational complexity in their effectiveness is a critical issue, there are in the literature analyses of this question from different perspectives (Echegoyen et al. 2011; Gao and Culberson 2005; Mühlenbein 2012).

In addition to sampling methods as implemented in EDAs, there are other ways to use probabilistic models to solve optimization problems. In this paper, we focus on a class of methods called message-passing algorithms (MPAs; Frey and Dueck 2007; Mézard et al. 2002; Pearl 1988; Yedidia et al. 2005) that exploit the problem decomposition explicitly captured in the model but organize the construction of the solutions in a different way to that used in EDAs. These algorithms work by exchanging messages between the components of the models until a solution is eventually reached. We claim that these methods, in combination with EDAs, have a good potential for the conception of more efficient hybrid optimization algorithms that incorporate, to a higher level, available machine learning strategies for problem decomposition.

Although a number of EDA papers have proposed strategies that adapt MPAs to evolutionary computation (Höns 2006; Lima et al. 2009; Mendiburu et al. 2007a, 2012; Soto 2003), the possibilities that these algorithms open for their use in EDAs are largely unexploited within the evolutionary computation community. This is particularly true for recent extensions and developments to MPAs. In this paper we provide arguments to illustrate how the application of MPAs will not only enhance the capacity of EDAs as problem solvers, but will also extend EDAs applicability to other areas such as the extraction of structural information from the problem domain.

The objective of this paper is threefold. First, we review and present a classification of different types of MPAs emphasizing on their dynamics and those characteristics that should be taken into account for their use in EDAs. Secondly, we review in detail previous applications of MPAs in EAs. Finally, we discuss a number of areas where the use of MPAs could lead to further improvements in the efficiency and scope of the applicability of EDAs.

2 EDAs

EDAs share with GAs the selection operator that selects the best solutions of the population according to its fitness. However, while GAs apply crossover and mutation operators to generate new solutions, in EDAs the new population is sampled from the joint probability distribution estimated from the selected individuals. Thus, the characteristic features of the best solutions in each generation are explicitly expressed through the joint probability distribution. A common pseudo-code for all EDAs is described in Algorithm 1.

The termination criteria of an EDA can be a maximum number of generations, a homogeneous population or no improvement after a specified number of generations. The probabilistic model learned at step 5 has a significant influence on the behavior of the EDA from the viewpoint of complexity and performance. EDAs mainly differ on the class of probabilistic model used to represent the distribution of the selected solutions.

Early work on EDAs (Mühlenbein and Mahnig 2002; Mühlenbein et al. 1999) has established that, for additively decomposable functions (ADFs) which are separable, if the ADF structure is accurately captured in the probabilistic model learned by the EDA, the algorithm will eventually converge to the optimum. Theoretical results in this direction were obtained for the Boltzmann estimation of distribution algorithm (Mühlenbein et al. 1999). However, learning an exact factorization of the problem structure is not necessary for an efficient search. Some problems can be solved with models that capture only an approximation of the problem structure (Brownlee et al. 2010; Echegoyen et al. 2007; Grahl et al. 2008).

Other components of the EDA have also an important influence in the way the structure of the problem is captured in the model. In particular, several papers have investigated the role of selection (Brownlee et al. 2008, 2012; González et al. 2002; Johnson and Shapiro 2002; Lima et al. 2007; Santana et al. 2005). It has been reported that in some functions, the structure of the problem learned by the models can be accurately recovered using a set of low-fitness solutions (bottom-selection; Brownlee et al. 2008). The study of the relationship between model building accuracy and optimization performance is an active research area in EDAs. Since MPAs can help to obtain the solutions with the highest probability given the model, they could shed some light on the question of whether more accurate models produce higher fitness solutions (Echegoyen et al. 2012).

3 Message passing algorithms

In this section we will illustrate the rationale of MPAs using a paradigmatic case of these methods: the belief propagation (BP) algorithm (Pearl 1988). BP is explained using a factor graph that will explicitly represent a factorization of the associated distribution and will also serve as an implicit model for problem decomposition.

3.1 Factor graphs

Let X be a random variable. A value of X is denoted x. $\mathbf{X} = (X_1, \ldots, X_n)$ will denote a vector of random variables. We will use $\mathbf{x} = (x_1, \ldots, x_n)$ to denote an assignment to the variables. S will denote a set of indexes in $N = \{1, \ldots, n\}$, and $\mathbf{X}_S$ (respectively, $\mathbf{x}_S$) a subset of the variables of $\mathbf{X}$ (respectively, a subset of values of $\mathbf{x}$) determined by the indices in S. $p(\mathbf{X} = \mathbf{x})$ is the joint probability mass function for $\mathbf{X}$. Similarly, $p(X_i = x_i)$ is the marginal mass probability function of X_i and $p(x_i \mid x_j)$ is the conditional mass probability of X_i given $X_j = x_j$.

A *factor graph* (Kschischang et al. 2001) is a bipartite graph that can serve to represent the factorized structure of a distribution. It has two types of nodes: variable nodes (represented as a circle), and factor nodes (represented as a square). Variable nodes usually represent single variables and factors represent a group of variables related by a function. In the graphs, factor nodes are named by capital letters starting from A, and variable nodes by numbers starting from 1. Figure 1 shows a factor graph with two factor nodes and five variable nodes. The existence of an edge connecting variable node i to factor node A means that x_i is an argument of function f_a in the referred factorization. The factorization represented by the factor graph in Fig. 1 is $g(x_1, x_2, x_3, x_4, x_5) = f_A(x_1, x_2, x_3)f_B(x_2, x_3, x_4, x_5)$.
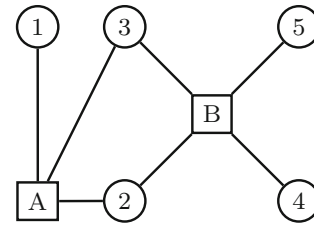

Fig. 1 Factor graph. Factor nodes are named by *capital letters* starting from A, and variable nodes by numbers starting from 1. In this example, there are two factor nodes: factor A, linked to variables X_1, X_2, X_3; and factor B linked to variables X_2, X_3, X_4, X_5

In Abbeel et al. (2006), Gibbs distributions are associated with factor graphs. A factor f with scope $\mathbf{X}_S$ is a mapping from $\mathbf{X}_S$ to $\mathbb{R}^+$. A Gibbs distribution $p(\mathbf{x})$ is associated with a set of factors $\{f_a\}_{a=1}^m$ with scopes $\{\mathbf{X}_{S_a}\}_{a=1}^m$, such that

$$p_f(\mathbf{x}) = \frac{1}{Z} \prod_{a=1}^m f_a(\mathbf{x}_{S_a}), \tag{1}$$

where $Z = \sum_{\mathbf{x}} \prod_{a=1}^m f_a(\mathbf{x}_{S_a})$ is the partition function.

Notice that the graphical representation of a factor graph can be also used to represent the relationship between the components of a problem and its variables. Even if we do not know the particular expression of the functions affecting each component, the mapping between the variables and problem components can contain valuable information for solving and understanding the problem.

3.2 Belief propagation

Given a factor graph, the most common inference problems are:

(1) *Marginal probabilities* to determine the marginal probability given by the model to a variable or set of variables $[p(\mathbf{x}_S)]$.

(2) *Most probable configuration* (*MPC*) to find the joint variables assignment that is given the highest probability by the factor graph, i.e., $[\mathbf{x}^* = arg\,max_{\mathbf{x}} p(\mathbf{x})]$.

BP is an iterative method that is used to solve these inference problems. The sum–product BP variant is used to compute the marginal probabilities, and the max-product BP is used to compute the MPC. The dynamics of the algorithms are similar for the two variants. However, each variant uses its own equations to update the messages.

When using BP with factor graphs, two kinds of messages are identified: messages $n_{i \to a}(x_i)$ sent from a variable node i to a factor node a, and messages $m_{a \to i}(x_i)$ sent from a factor node a to a variable node i. Note that a message is sent for every value of each variable X_i.

These messages are updated according to the following rules:

$$n_{i \to a}(x_i) := \prod_{c \in N(i) \backslash a} m_{c \to i}(x_i), \qquad (2)$$

$$m_{a \to i}(x_i) := \sum_{x_a \backslash x_i} f_a(x_a) \prod_{j \in N(a) \backslash i} n_{j \to a}(x_j), \qquad (3)$$

$$m_{a \to i}(x_i) := \arg \max_{x_a \backslash x_i} \left\{ f_a(x_a) \prod_{j \in N(a) \backslash i} n_{j \to a}(x_j) \right\}, \qquad (4)$$

where $N(i) \backslash a$ represents all the neighboring factor nodes of node i excluding node a, and $\sum_{x_a \backslash x_i}$ expresses that the sum is completed taking into account all the possible values that all variables in X_a except X_i can take—while variable X_i takes its x_i value.

Equations 2 and 3 are used when marginal probabilities are searched for (sum–product). By contrast, in order to obtain the MPC (max-product), Eqs. 2 and 4 should be applied.

When the algorithm converges (i.e., messages do not change), marginal functions (sum–product) or max-marginals (max-product) are obtained as the normalized product of all messages received by X_i :

$$g_i(x_i) \propto \prod_{a \in N(i)} m_{a \to i}(x_i). \qquad (5)$$

Regarding the max-product approach, once the algorithm has converged, the optimal solution is obtained by combining the assignments with the highest probability in each max-marginal.

The pseudocode of a generic BP algorithm on factor graphs is shown in Algorithm 2. Two important elements that influence the algorithm are the order of the updates of the nodes (scheduling) and the termination criterion. When the factor graph is acyclic, an optimal scheduling can be found. The graph is organized as a tree, and the root and leaves are identified. Messages are propagated from the leaves toward the root node and then from the root toward the leaves. Implementation of BP and other MPAs can be performed according synchronous or asynchronous schemes that specify the time in which messages are updated and may determine different convergence properties of the algorithm (Crick and Pfeffer 2003).

One possible termination criterion can be the convergence of the algorithm. For example, that at some iteration t $|m_{a \to i}^t - m_{a \to i}^{t-1}| < \alpha$ for each edge of the graph. Another common termination criterion is that a maximum number of iterations has been reached. More details on the message passing strategies used on factor graphs can be found in Höns (2006), Kschischang et al. (2001) and Yedidia et al. (2005).

One characteristic feature in all BP implementations is an iterative and distributed process in which the nodes of the graph receive and send messages containing information about the probabilities of each of the possible factor configurations. Using a metaphor from GA schema theory (Goldberg 1989), it is as if a predefined set of schemata, represented by the factors, exchange information about the likelihood of each of their possible states, trying to find the optimal configuration for all the schemata. If we see the formation of optimal solutions in EAs as a vertical process in which each generation corresponds to a different level, and solutions are progressively conformed and refined from one level to the next, then we can see MPAs as a horizontal process in which the consensus about the problem solution is reached horizontally, in one single level and based on the computation of the messages.

One of the contributions of MPAs–EAs is therefore to provide a different way to reach the optimal solutions and one of its potential applications in EAs lies in the multiple forms in which the vertical and horizontal processes for creating new solutions can be combined.

4 Classification of MPAs

BP is the simplest MPA. Several variants have been developed and the rationale behind these developments is often relevant for potential applications in EDAs. We focus on three main classes of MPAs that differ in their goals and in the way message propagation is conducted: (1) extensions to the simple BP, (2) survey propagation (SP) and warning propagation (WP) algorithms, and (3) affinity propagation (AP).

Algorithm 2 Belief propagation on factor graphs

1   For $t = 0$, initialize the messages $n_{i \to a}$ and $m_{a \to i}$ along the edges of the factor graph

2   Determine a sequential order for the edges in the factor graph

3   **do** {

4   Update sequentially the messages on all edges of the graph, generating new set of messages $n_{i \to a}$ and $m_{a \to i}$, according to message update rules in Equations (2) and (3), respectively.

5   } **until** Termination criterion is met

4.1 Extensions to BP

The main drawback of BP and other MPAs is related to its convergence properties. Although there are results that give necessary conditions for convergence (Heskes 2004), the algorithm is not guaranteed to converge for every problem. There are many issues that influence BP convergence. The topology of the graph, which is strongly related to the problem structure, is one of the crucial elements. Even if the algorithm converges, difficulties in obtaining the solution with the highest probability may arise when there are multiple local optimal solutions in the factors (Meltzer et al. 2005). Thus, extensions to BP have tried to improve these results by, for example, redefining the initial graphs where MPA is applied (Table 1).

- *Loopy BP* (*LBP*) BP is guaranteed to converge in graphs without cycles. Therefore, the straightforward extension of BP is its application to loopy graphs where the algorithm is not guaranteed to converge (Pearl 1988). LBP has been extensively applied to many different domains and theoretical and empirical results have investigated different facets of its dynamics (Ihler et al. 2006; Murphy et al. 1999; Weiss and Freeman 2001). The way the algorithm is implemented may be critical for its application. Messages can be updated sequentially or in parallel. The fixed points of the algorithm do not change as a result of the schedule, but the dynamics of the algorithm does. This fact is relevant when considering uses of MPAs and their implementations.

Table 1 Description of surveyed message-passing algorithms

Algorithms	References	Discrete	Continue
BP	Bickson (2008) and Pearl (1988)	x	x
Loopy BP	Bickson (2008) and Pearl (1988)	x	x
Generalized BP	Tanaka et al. (2004) and Yedidia et al. (2005)	x	x
Expectation propagation	Minka (2001)	x	x
Fractional BP	Wiegerinck and Heskes (2003)	x	
Tree-reweighted BP	Wainwright et al. (2004)	x	
Concave–convex procedure	Yuille (2001)	x	
Warning propagation	Mézard et al. (2002)	x	
Survey propagation	Mézard et al. (2002)	x	
Affinity propagation	Frey and Dueck (2007)	x	x

- *Generalized BP* (*GBP*; Yedidia et al. 2005) the key idea of GBP is the definition of regions that can cover more than one factor of the original factor graph. This type of region based decompositions are commonly used in statistical physics to obtain better approximations of the free energy and the entropy. GBP typically allows to achieve more accurate results and better convergence properties than BP (Domínguez et al. 2011; Yanover and Weiss 2003; Yedidia et al. 2005).
- *Expectation propagation* (*EP*; Minka 2001) EP approximates the belief states by only retaining expectations, such as mean and variance, and iterates until these expectations are consistent throughout the graphical model. This makes it applicable to hybrid networks with discrete and continuous nodes.
- *Fractional BP* (*FBP*; Wiegerinck and Heskes 2003) the idea of FBP is to deal with the cycles that affect BP convergence without resorting to the large regions GBP uses. In FBP, the approximations are parametrized by scale parameters in order to better model the effective interactions due to the effect of loops in the graph. FBP represents another alternative for problems with loopy interactions.
- *Concave–convex procedure* (*CCCP*; Yuille 2001) CCCP also propagates messages (formally Lagrange multipliers), but unlike BP and GBP, the propagation depends on current estimates of the beliefs that must be re-estimated periodically. CCCP consists of an inner loop in which the messages are updated until convergence, and an outer loop in which the current estimates of the beliefs are updated.
- *Tree-reweighted re-parametrization* (*TRW*; Wainwright et al. 2004) TRW is based on the idea of re-parametrizing the distribution in terms of so-called pseudo-max-marginals on nodes and edges of the graph. The ultimate goal is to obtain a re-parametrization for which tree-consistency holds for every spanning tree of the graph simultaneously. In TRW, the structure of the factor graph is decomposed by a set of overlapping spanning hypertrees from which junction-tree-factorized distributions can be computed.

There are other message-passing-scheme related MPAs. Some implementations of iterative proportional fitting (IPF; Deming and Stephan 1940) comprise a message passing step similar to the one used by BP (Jiroušek and Přeučil 1995). However, since the goal of IPF is to find the maximum entropy distribution given a set of marginal probabilities, we do not consider this type of MPAs in detail. Decimation strategies are another class of technique recently applied together with MPAs for max-propagation. They work by repeatedly fixing variable values and simplifying without reconsidering earlier decisions (Kroc et al.

2009). When, along the search for a maximum configuration, there is certainty about the values of some variables, this information can be used to simplify the problem and speed convergence.

4.2 WP and SP algorithms

WP and SP (Mézard et al. 2002) offer a completely different perspective to the use and insights provided by MPAs–EDAs. The reason is that they were conceived for the solution of combinatorial optimization problems such as satisfiability (SAT), minimum vertex cover, etc. Some of the theoretical analysis explaining the behavior of these algorithms is relevant to understand the limits of EDAs in their application to combinatorial problems.

4.2.1 WP algorithm

We use the SAT problem to illustrate WP. For this problem, every clause is taken as a factor of the factor graph. For every variable node i in this graph, we denote by $V(i)$ the set of function nodes a to which it is connected by an edge, by $V(i)_+$ the subset of function nodes a where the variable appears un-negated, and by $V(i)_-$ the complementary subset of $V(i)$ consisting of function nodes a where the variable appears negated.

The basic elementary message passed from one function node a to a variable i is a Boolean number $u_{a \to i} \in \{0, 1\}$ called a *cavity bias*. The basic elementary message passed from one variable node j to a function node a is an integer number $h_{j \to a}$ called a *cavity field*.

A cavity bias $u_{a \to i} = 1$ can be interpreted as a warning sent from function node a, telling variable i that it should adopt the correct value for satisfying the clause a. If $h_{j \to a} > 0$, this means that the tendency for site j (in the absence of a) would be to take a value which does not satisfy clause a.

Similarly to the BP method, in order to compute the cavity bias, the function node a considers the incoming cavity fields which it receives from all the variable nodes j to which it is connected to.

WP receives as a parameter the maximal number of iterations t_{max}. If after this number of iterations the algorithm has failed to converge, it halts. Otherwise it outputs a fixed set of cavity biases $u^*_{a \to i}$ from which the local fields H_I and the contradiction number c_i can be computed.

The local field is an indication of the preferred state of the variable i: $x_i = 1$ if $H_i > 0$, $x_i = 0$ if $H_i < 0$. The contradiction number indicates whether the variable i has received conflicting messages.

WP illustrates to a greater extent than BP how message-passing is instrumental in the construction of the optimal solutions (in this case, optimal SAT assignments).

4.2.2 SP algorithm

SP is another MPA that has been successfully applied to the solution of random 3-SAT problems sampled from the hard part of the SAT phase (the transition region; Braunstein et al. 2005; Mézard et al. 2002).

The difficulty posed by instances in the transition region of SAT is due to a number of characteristics also shared by hard instances of other combinatorial problems. These are:

– The solution space breaks up into many clusters.
– Solutions in separate clusters are generally far apart.
– Clusters that correspond to partial solutions are exponentially more numerous than the clusters of complete solutions.

The existence of such a clustering of the search space, and the fluctuation that the values of the variables experience at the different clusters determine that WP and BP fail to converge. SP has been conceived to address these situations. In SP, the messages are surveys of some other elementary (warning) messages which are probability distributions parametrized in a simple way. These probability distributions describe how the single Boolean variables are expected to fluctuate from cluster to cluster.

A *survey* of a cavity bias $u_{a \to i}$ is defined as:

$$Q_{a \to i}(u) = \frac{1}{\mathcal{N}_{cl}} \sum_l \delta\big(u, u^l_{a \to i}\big), \tag{6}$$

where l runs over all clusters of solutions and $u^l_{a \to i}$ is the cavity bias over $a \to i$ in the cluster l. δ denotes the Kronecker delta function over discrete variables and $\mathcal{N}_{cl}$ is the estimated number of clusters. Since $u \in \{0, 1\}$, the cavity-bias-survey $Q_{a \to i}(u)$, which is a probability distribution, is parametrized by a single number.

$$Q_{a \to i}(u) = (1 - \eta_{a \to i})\delta(u, 0) + \eta_{a \to i}\delta(u, 1). \tag{7}$$

The SP iteration equations can be defined in terms of the η variables.

There are fundamental differences between SP and BP algorithms. These differences are reviewed in Braunstein et al. (2005). In comparison to BP, SP introduces a new state of the variables, a "joker state", that means that variables are not constrained. Another convenient feature of SP is that valuable information about the problem structure can be extracted from the set of surveys.

SP has also been applied to the vertex cover problem and generalized to other combinatorial problems with constraints (Braunstein et al. 2006). The application of SP to the vertex cover problem has outperformed local heuristics on random graphs of high but finite average vertex degree, for which WP and BP stop converging (Hartmann and Weigt 2005).

SP is an example of possible alternatives to deal with the basic incapability of elementary MPAs to describe the clustering phenomenon. From this perspective, it can be seen as an improved version of BP algorithms. It is also at the core of a problem simplification technique for solving combinatorial problems. The survey inspired decimation technique (Mézard et al. 2002) is a relevant example of the ways in which MPAs can be inserted in more general optimization strategies.

4.3 Affinity propagation

Finally, we describe another MPA whose domain of application is different to BP, SW, and WP. AP (Frey and Dueck 2007) is a message-passing-based clustering algorithm that, given a set of points and a set of similarity values between the points, finds clusters of similar points, and for each cluster gives a representative example or exemplar.

Here the role of MPA is not to find a distribution or to construct the optimal solution to a combinatorial problem. However, the strategy of exchanging messages as a way to share and condense local information with the aim of solving a global problem is also applied and reports similar benefits. Since several tasks in EDAs and other EAs can be posed as a clustering problem, AP gives another perspective of MPA applicability.

AP takes as input a matrix of similarity measures between each pair of points $s(\mathbf{y}^i, \mathbf{y}^k)$. For each data point $\mathbf{y}^k$, a real number $s(\mathbf{y}^k, \mathbf{y}^k)$ is also entered as an initial input. The $s(\mathbf{y}^k, \mathbf{y}^k)$ values are called *preferences* and are a measure of how likely each point is to be chosen as exemplar. The algorithm works by exchanging messages between the points until a stop condition, which reflects an agreement between all the points with respect to the current assignment of the exemplars, is reached. These messages can be seen as the way the points share local information in the gradual determination of the exemplars.

Preferences can be tuned to influence the number and compositions of the clusters and to introduce a priori information about which are the points that best represent the different type of data. Several improvements and variants of AP have been proposed (Givoni and Frey 2009; Givoni et al. 2011; Leone et al. 2007).

5 Applications of MPAs in EDAs

5.1 Applications of BP and its generalizations

One of the first applications of MPAs in EDAs was presented in Soto (2003), where Nilsson's algorithm (1998) to find the MPC in a junction tree was used in the context of optimization by EDAs. The algorithm was applied to obtain the MPCs of EDAs based on trees and polytrees. Results on the influence of computing the MPA were also presented for an EDA that uses a univariate model. For this type of model the application of LBP is not required in order to find the MPC. Since for this model variables are considered to be independent, the MPC is formed by those configurations for which each univariate probability is maximized. For trees and polytrees, the application of LBP is required and it provably converges to the MPC. The results reported in Soto (2003) proved that EDAs that combine probabilistic logic sampling (PLS; Henrion 1988) with the computation of the MPC were more efficient in terms of function evaluations than EDAs that use only PLS.

In Santana (2006), the use of LBP in EDAs was extended from tree models to pairwise Markov networks. Two EDA variants that incorporate BP were applied to a protein design problem for which the (loopy) structure of the interactions was known a priori. The first variant, EDA-Loopy-P, samples the new population of solutions applying PLS to a tree learned from the selected solutions. The MPC is computed from the graphical model and added to the new population. The graphical model structure used by LBP is determined by the a priori known structure of the protein while its marginals are learned from data. The second variant, EDA-Loopy, is similar to EDA-Loopy-P but only interactions known a priori can be included in the tree model learned from data. Results shown in Santana (2006) revealed that EDA-Loopy-P outperformed an EDA that exclusively employs PLS. However, the use of LBP did not improve EDAs that fully incorporate a priori information about the problem structure.

In Mendiburu et al. (2007b, 2012), LBP was applied to compute the MPCs in the context of optimization with estimation of Bayesian network algorithms (EBNAs; Etxeberria and Larrañaga 1999). This proposal incorporates an intermediate step in which the Bayesian network learned from data was transformed to a factor graph. Experiments were carried out in functions comprising different sources of difficulty for EAs. Although the experimental results showed no significant differences in the fitness of the best solutions due to the use of LBP, an analysis of the EDA behaviors showed that LBP clearly helps EBNA to obtain better individuals in earlier generations. Another difference identified between the behavior of EBNA and EBNA–LBP is that the second algorithm produced a higher deviation of the mean best fitness values.

A number of important questions about the relationship between the problem structure, the probabilistic models learned by EDAs, and the quality of the sampled solutions were analyzed in Echegoyen et al. (2010a). In this paper, the evolution of the probabilities given by the model to the best solution and to the MPC were investigated in relationship to the type and amount of dependencies learned by

the model. BP was applied in junction trees to learn the MPC that was added to the new population. Experimental results showed that the fitness function corresponding to the MPC always increases along the generations of the EDA (Echegoyen et al. 2010a). Moreover, it usually has a higher fitness function than the best individual of the population, particularly in the early generations. This result agrees with previous reports on the evolution of the MPC probability (Mendiburu et al. 2007b, 2012).

Two critical questions for the application of MPAs in EDAs are to have a flexible implementation and to be scalable. Flexibility is needed for allowing the manipulation of the MPA components (i.e., initialization methods, scheduling policies, and stopping criteria) according to the search goals, and for the combination of these components with the probabilistic model learning and sampling steps used by EDAs. Scalability is essential for addressing hard optimization problems with many variables in a reasonable time. In Mendiburu et al. (2007a), a flexible parallel framework for LBP over factor graphs is introduced. Parallelization is shown to be a suitable alternative for achieving more efficient LBP implementations. Parallelization has produced important improvements over serial implementations (Mendiburu et al. 2005) in EDAs. Similarly, the work presented in Mendiburu et al. (2007a) shows that the optimal solution of the optimization problems can be found more efficiently by using parallel implementations of BP.

Different papers apply efficient IPF implementations, that include a message-passing step, to compute maximum-entropy approximations of the selected population in EDAs (Höns 2006; Ochoa et al. 2003). This approach is followed in Ochoa et al. (2003) to reduce the population size needed by the polytree approximation distribution algorithm based on second order marginals (PADA2). A problem of PADA2 is that while it needs only bivariate distributions for constructing the polytree, the marginals needed for sampling are larger. Therefore, in Ochoa et al. (2003) low order marginals learned from the data are combined in cliques of higher size by applying IPF on a junction tree structure where messages are passed between the cliques in the tree. The same idea is extended in Höns (2006) to learn probability approximations in EDAs that use generalized factorizations. The maximum entropy FDA (MEFDA) computes the maximum entropy distribution from smaller marginals learned from the data. Similarly to the results achieved for PADA2, MEFDA is able to reduce the population size needed to optimize problems of different complexity.

In Lima (2009) and Lima et al. (2009), LBP was applied to perform a substructural local search (SLS), looking for the solution with the highest fitness value instead of that with the highest probability. Authors conclude that Loopy-SLS is more effective than other SLS methods for problems with substantial overlap between subproblems for which an exhaustive analysis and combination of the local configurations corresponding to each marginal is very costly. The approach is similar to the one used in Mendiburu et al. (2005) for finding the MPC. However, in Lima (2009) and Lima et al. (2009) the factor nodes use fitness information instead of the traditional approach in LBP which is to use the marginal probabilities. The experimental results on deceptive functions showed that Loopy-SLS can take more advantage from larger populations to reduce the number of function evaluations in comparison with the classical LBP approach. More accurate fitness information allows the loopy local searcher to converge faster to optimal solutions (Lima 2009).

More recently, a number of methods for generating new solutions in EDAs that use Markov networks have been proposed (Santana et al. 2012b, 2013). These methods intend to diminish the computational cost associated to the sampling process of the Markov optimization algorithm (MOA; Shakya et al. 2012) and the Markov network FDA based on G tests (MN-FDA^G; Santana et al. 2012b). These EDAs compute the MPC using LBP and decimation-BP (Kroc et al. 2009; Mooij 2010). The MPC can be used in two forms: adding it to the new generated population, or as a template for crossover. When the MPC is used as a template, it serves to recombine the information of the solutions in the current population with the global information contained in the Markov model. Recombination guarantees that partial configurations which coincide with the MPC will remain in the next population. These methods were compared to several other algorithms used to generate new solutions in discrete functions of different cardinalities. One of the conclusions of the experimental results is that adding the MPC has a positive effect in terms of the number of function evaluations. Nevertheless, in several cases the gain will depend on the combination of the MPC with other generation techniques.

In a series of papers (Höns 2006, 2012; Mühlenbein and Höns 2006) examples in which MPAs are used in their sum–product version were presented. These papers presented different variants of the Bethe–Kikuchi distribution algorithm (BKDA), an EDA whose general goal is to learn Bethe–Kikuchi approximations of the Boltzmann distribution. Given a selected set of solutions and an arbitrary factorization, the BKDA uses the marginals of the factorization to create a region graph, then a MPA is applied to compute the set of consistent marginals, and after these marginals have been computed the FDA factorization is used again for sampling. In Höns (2012), the MPA of choice is the GBP, which is tested using different types of factorizations for the Ising model. In Mühlenbein and Höns (2006), CCCP, which alternates between updates of the

convex and concave term of the Lagrangian that enforces the consistency constraints, is used instead. CCCP can deal with lack of GBP convergence due to numerical instability. Results of BKDA for Ising and Kauffman's $n - k$ functions were not clearly superior to results achieved with the FDA, however the former algorithm required a population size three times lower than the second for most of the problems, and BKDA runs slightly faster than FDA. Details of the two BKDA variants were presented in Höns (2006).

In Höns et al. (2007), max-GBP is used to compute an approximation of the k MPCs. The loopy max-flow propagation algorithm (LMFP) uses the same partition proposed in Nilsson (1998) to find the k MPCs of a junction tree but instead of applying BP, it uses GBP in region graphs corresponding to different factorizations. LMFP is compared to sampling methods using sum-GBP in the context of BKDA. The results showed that in difficult Ising instances, the sum-GBP produces a low probability of the optimum, whereas max-GBP does not converge and finds suboptimal solutions.

The application of MPAs to genetic programming has been recently proposed (Sato et al. 2012). BP is used in the context of probabilistic modeling of prototype tree-based methods. The program optimization with linkage estimation (POLE) employs Bayesian networks for a probabilistic model and uses a special chromosome called the expanded parse tree. Its hybrid variant (POLE-BP) uses BP to compute the MPC of the probabilistic model. The empirical results presented in Sato et al. (2012) show that POLE-BP works better than POLE in deeper problems which require a larger population size.

5.2 Applications of SP

In comparison to the use of BP and its extended variants in EAs, the use of WP, SP and AP has been more scarce. Recently, Furtlehner and Schoenauer (2010) proposed the use of SP to sample the Pareto set of solutions in a bi-objective problem. Although the algorithm can not be considered as an EA, the use of sampling methods inspired by decimation strategies as those used with SP point to a promising use of SP within EAs.

5.3 Applications of AP

In the context of evolutionary computation, AP has been used as a way to identify clusters of related variables in discrete and continuous problems. In this type of application, a measure of interactions between the variables is used as the initial similarity matrix. For discrete problems, the mutual information has been used in Santana et al. (2010) and the covariance matrix in Karshenas et al. (2012). The output of AP is a set of factors, each factor containing the set of interacting variables. A modification to the original AP has been introduced in Santana et al. (2010) to guarantee that all factors have a constrained size. This is required for allowing feasible marginal product models (Harik 1999). The experimental results of the AP–EDA presented in Santana et al. (2010) showed that the algorithm was able to outperform other EDAs based on MPMs such as the extended compact GA (Harik 1999) for a set of binary and non-binary discrete deceptive functions.

AP has also been used as a niching procedure for EDAs based on Markov chain models (Chen and Hu 2010, 2010b). In this case, instead of clustering the variables according to the strength of the interactions, solutions are clustered and a different probabilistic model is learned in each cluster. The algorithm was applied to a simplified protein model and compared with other EDAs and other classes of optimization methods. The experimental results presented in Chen and Hu (2010) showed that an EDA enhanced by AP has a greater chance to find the global optimum or good suboptimal solutions for protein sequences than the two other simpler EDA variants compared. It also achieved very competitive results with respect to other optimization algorithms.

6 Towards a better synergy between MPAs and EDAs

In this section we analyze some possible synergies between MPAs and EDAs that can result in the design of more effective and efficient optimization algorithms. We present the potential application of MPAs–EDAs, explaining the rationale behind our proposal and preliminary results that can support our statements. We also analyze current research topics on the integration between MPAs and EDAs and open problems whose answers could advance the applicability of hybrid MPAs–EDAs approaches (Table 2).

6.1 Effective and feasible factorizations

A fundamental and scarcely investigated issue in MPAs is how the choice of the initial regions influences the quality of the approximations and convergence properties of MPAs. Graph partitioning algorithms have been proposed to determine the set of initial clusters in the context of generalized mean field algorithms (Xing and Jordan 2003). Sequential strategies have been also proposed (Welling 2004) as a way to set the initial regions for GBP. In EDAs, the choice of the selected regions is linked to the accuracy and the cost of the factorization. Höns (2006) and Höns (2012) has proposed using the regular structure of the Ising problem to construct more comprehensive region-based decompositions that serve as a basis for pentavariate

Table 2 Application of message-passing algorithms to EDAs

EDA	MPA	Types	References
EDA-Loopy-P	LBP	Max	Santana (2006)
EBNA-BP	LBP	Max	Mendiburu et al. (2007b, 2012)
PADA	BP	–	Höns et al. (2007) and Ochoa et al. (2003)
EBNA	BP	Max	Echegoyen (2012) and Echegoyen et al. (2009, 2010a)
BOA	LBP	Max	Lima (2009) and Lima et al. (2009)
MOA, MN-FDA	LBP	Max	Santana et al. (2012b)
MOA, MN-FDA	Decimation-BP	Max	Santana et al. (2012b)
POLE-BP	LBP	Max	Sato et al. (2012)
PADA2	IPF	–	Ochoa et al. (2003)
MEFDA	IPF	Sum	Höns (2006)
BKDA	CCCP	Sum	Höns (2006) and Mühlenbein and Höns (2006)
BKDA	GBP	Sum	Höns (2006, 2012)
BKDA	GBP	Max	Höns et al. (2007)
BKDA	CCCP	Max	Höns (2012)
MOEA	SP	–	Furtlehner and Schoenauer (2010)
AffEDA	AP	–	Santana et al. (2010)
MK-EDA	AP	–	Chen and Hu (2010b, 2010)
GMRF-EDA	AP	–	Karshenas et al. (2012)

factorizations. However, the question of how to deal with problems with irregular structure is still open. This is a research area where new breakthroughs could benefit MPAs and EDAs.

6.2 Model learning and decompositions

The application of MPAs to model learning can be extended beyond the use of AP for MPMs in which factors do not overlap to the construction of factorized models with overlapped variables. Recent versions of AP (Givoni et al. 2011; Leone et al. 2007) allow to obtain overlapping clusters opening the possibility to create more general model-learning algorithms based on AP.

One challenge for EDAs and other EAs are optimization problems with hundreds or thousands of variables. Although approaches that consider the independence between the variables have been applied (Sastry et al. 2007; Suwannik and Chongstitvatana 2008), other EDA traditional model learning approaches may require population sizes that are not affordable from a computational point of view. One solution is the conception of decomposition strategies that implement learning of the simpler probabilistic models in parallel and exchange the information between the models using MPAs. Another feasible alternative is the use of dimensionality reduction techniques. Two examples of this type of techniques that have been successfully applied in EAs are random projections (Kaban et al. 2013; Wang et al. 2013) and non-negative factorization (Helmi et al. 2014).

Models of different complexity can be also associated to different sets of variables. Model complexity could be allowed to be higher for those subsets of variables that are known in advance to have stronger interactions. Another possibility is to use the information available a priori about the interactions between the variables to propose simplifications or decompositions of the original graph that could serve for parallelization. One example of these decomposition strategies used in MPAs is the TRBP algorithm in which the original problem on the graph with cycles is represented as a convex combination of tree-structured problems (Wainwright et al. 2002). A variety of extensions to tree factorizations have been proposed in EDAs (Baluja and Davies 1997; Kim et al. 2010; Ponce de León and Díaz 2012). However, none of them applies the idea of simultaneously using different spanning trees of the same model.

The use of MPAs for model decomposition could allow to exchange information between the model components or apply different techniques for generating partial solutions (e.g., PLS and MPA). Potential decomposition approaches in EDAs can also benefit from MPA decomposition strategies that consider subgraphs more complex than trees (Batra et al. 2010) and from hybrid strategies that combine MPAs with other combinatorial algorithms (Duchi et al. 2007).

6.3 Synchronization problem

In GAs, the synchronization problem occurs when the algorithm gets stuck in a local optimum because the schemata in its current population, which are non-inferior

building blocks of different optima, cannot be combined to form an optimum (Van Hoyweghen et al. 2002a, b).

In MPAs a similar issue arises. There are situations where the MPA has converged, meaning that all max-marginals are locally consistent but not necessarily implying that globally consistency has been achieved (Wainwright and Jordan 2003). The problem arises when there are regions for which the maximal belief is reached at multiple configurations. Therefore, if the maxima in the single nodes are not unique, it might be impossible to combine them to a consistent maximum (Wainwright et al. 2004). The existence of ties within nodes is an element of difficulty for other MPAs such as TRBP (Meltzer et al. 2005).

The combination of EDAs with MPAs provide, on one hand, the possibility of identifying the sets of interacting variables by means of statistical tests applied on the data. On the other hand, the capacity to identify all local configurations that maximize the probability of interacting variables by means of MPAs. The problem that remains then is how to efficiently combine all possible locally optimal configurations to obtain the global solutions.

6.4 Multiple local optima and symmetry breaking

Problems with many clusters of local solutions (e.g., hard SAT instances Braunstein et al. 2005; Mézard et al. 2002) are not easier for EDAs and EAs than for BP, WP, and other MPAs. Therefore, there is a profuse amount of literature on EA methods to enforce the diversity of the solutions and search for the optima in hard multi-modal problems. One paradigmatic example of these approaches is niching methods (Mahfoud 1995). These methods work by creating and keeping stable subpopulations around different local optima. Although niching methods have been combined with different variants of EDAs (Dong and Yao 2008; Sastry et al. 2005), more specific EDA approaches to multi-modal optima are the application of probabilistic clustering algorithms and mixtures of distributions (Peña et al. 2005).

In terms of probabilistic modeling, the idea of identifying context-dependent probabilistic dependencies is not only an alternative to deal with multi-modal problems but also a more comprehensive approach to extract the structure and regularity patterns from these problems.

In addition to the application of MPAs for clustering of the solutions (Chen and Hu 2010, 2010b), WP and SP are promising alternatives to combine with EDAs. These MPAs provide an automatic method to extract precise statistical information of the relationship between local optima and variability of the configurations of the variables according to the clusters of the local optima. Decimation methods used with WP and SP can be also applied in EDAs.

6.5 Fitness modeling

One of the advantages of EDAs over GAs is that the probabilistic model can be used to approximate the fitness function or to serve as a fitness estimation at the time of comparing or ranking solutions. A variety of methods have been proposed for fitness estimation in EDAs (Brownlee 2009; Brownlee et al. 2008; Rivera and Santana 2000; Sastry et al. 2006).

MPAs can be used for fitness modeling in EDAs in multiple ways. One possibility is to add fitness information to the model and apply MPAs to obtain the MPA configuration, and its associated fitness estimation. This approach can be further extended to compute the k solutions with the highest estimated fitness according to the model. The generation of the k MPCs can be conducted in such a way that it could be possible to generate all solutions with a fitness above a given estimated value. However, such schemes can be very costly (Höns et al. 2007). More importantly, MPAs allow us to estimate the fitness of solutions for which a number of variables have been instantiated (i.e., evidence about some of the variables is introduced to the model and propagated).

Another possibility is to combine in the probabilistic graphical model (e.g., factor graph) diverse information of the optimization problem and propagate this information using MPAs. Such information can come from sampled solutions, known constraints, or other knowledge available about the way the function is decomposed. Related approaches are those used by EBNA–LBP and Loopy-SLP. While the former applies information learned from the data, the second incorporates information about the fitness. Notice however, that the information about the fitness can be explicitly included in the model by learning the marginal probabilities directly from the selection probabilities (Munetomo et al. 2008; Santana 2003; Valdez-Peña et al. 2009).

Finally, the application of MPAs within the EDA framework for modeling the fitness could be extended to multi-objective problems. For these problems, the exploitation of the relationships between the variables and the objectives as captured by the probabilistic models has been proposed with good empirical results (Karshenas et al. 2011). MPAs could be applied in a model that includes objectives and variables to combine information from the different objective functions.

6.6 Hybrid sampling strategies

Previous work on the use of the MPCs obtained using MPAs (Echegoyen et al. 2010a; Höns et al. 2007; Mendiburu et al. 2007b, 2012; Santana et al. 2012b) have shown that the addition of the MPCs tends to produce an

intensification effect on the search while the application of PLS serves to sample unexplored areas of the search space. Finding an adequate and dynamic balance between the use of these two methods would contribute to improve the efficiency of EDA sampling algorithms. Given the capacity of EDAs enhanced by MPAs to generate very good solutions from the beginning of the evolution, they could be used as a basis for the implementation of anytime algorithms (Zilberstein 1996). It is also possible to design hybrid partial sampling strategies in which some factors are sampled using PLS and other factors are sampled using the MPCs obtained from MPAs.

The application of WP and SP can be used as a step previous to the application of the EDA. If the MPA does not converge, the information about the preferred states of the variables or the cavity bias can be used as a way to sample an initial population of solutions. Variables that have not received conflicting messages will be strongly biased to the values found by the MPA. Variables that have received conflicting messages would be sampled more uniformly.

One of the current limitations of EDAs is in dealing with constraints. Probabilistic models are in general not suitable to represent general constraints between the variables and repairing procedures tend to disrupt the dependencies that have been captured and sampled by the probabilistic models. Constraints that involve a small number of variables can be represented by factors that, once included in the factor graph, can be optimized using EDAs that employ MPAs for sampling in an attempt to incorporate the constraints to the message-passing procedure.

6.7 Use of the topological features

For optimization problems for which the structure is known, it is usually possible to make a characterization of the problem based on topological features of the graph. It has been confirmed that some characteristics of the difficulty of a problem instance can also be captured by topological descriptors (Echegoyen 2012; Echegoyen et al. 2010b; Mooij 2005; Santana et al. 2008; Weiss 2000). In MPAs, the number and distribution of loops in the graph is a clear example of these topological characteristics (Weiss 2000). Loops can contribute to the arousal of frustration, although for this to happen the problem parameters have also a role to play. In EDAs, the influence of topological characteristics have also been investigated (Echegoyen 2012; Echegoyen et al. 2010b; Santana et al. 2008, 2012a). Furthermore, it has been confirmed that EDAs are able to capture structural properties of the optimization problem in the graphical structure of the probabilistic model (Brownlee et al. 2012; Echegoyen 2012; Echegoyen et al. 2008; Karshenas et al. 2011).

One open question is how to use the topological information of the problem to decide on the application of MPAs, EDAs, or any other possible combination of them. Analyzing whether problems defined on graph topologies for which MPAs have been theoretically proved to converge are equally solvable by EDAs is another interesting research topic. Results achieved for MPAs can provide some clues of the class of problems where EDAs are expected to be more successful. It is also an open question whether characterizations of the graphs in terms of topological measures can serve to predict the EDA behavior in these problems. Empirical analysis for the study of these measures on LBP have been conducted (Santana et al. 2008).

6.8 Dealing with continuous, permutation-based, and mixed problems

Although MPAs such as Gaussian BP have been proposed for continuous problems (Dolev et al. 2009; Malioutov et al. 2006), this type of MPAs have not been applied in EDAs. However, they represent an opportunity for continuous problems where EDAs usually require sophisticated mechanisms to avoid getting trapped in local optima. The application of EDAs to mixed problems, with continuous and discrete variables is also very limited by the complexity of the probabilistic models needed to represent these problem domains (Ocenasek and Schwarz 2002). MPAs such as EP (Minka 2001) show that in these models, inference methods based on MPAs are also possible.

Recent results on the application of probabilistic models defined on permutations (Ceberio et al. 2012, 2013), and the relatively simple way to compute the MPC for this type of models, point to the potential use of hybrid sampling methods for this domain. Preliminary results have already shown the feasibility of using some probabilistic models defined on permutations as fitness surrogates (Regnier-Coudert 2013). More work is needed in this direction.

6.9 Divergence measures

Recently, BP and mean-field algorithms have been analyzed from the perspective of the divergence measure they minimize (Minka 2005). The difference is stated not to be the amount of structure they model, but only the measure of loss they minimize. Minka (2005) identifies as an open research trend the construction of information divergence interpretation of other MPAs such as GBP and max-product versions of BP and TRW. Since the analysis of probabilistic approximations used by EDAs is usually focused only on the amount of dependencies represented by the models, this perspective is relevant to widen the scope of analysis of probabilistic approximations.

7 Conclusions

In this paper we have conducted an analysis of MPAs and their application within EDAs. Although there are several works that propose the use of MPAs in the context of EDAs, we consider that the potential for MPA applications in EDAs is far from being exhausted. To illustrate this point, we have reviewed a number of extensions to simple MPAs and discussed different ways these extensions could guide research on EDAs. Furthermore, there is also the possibility of using EDAs as a way to extend current MPAs, particularly for black-box optimization problems for which a description of the structural relationship among the variables is not available. By summarizing current work on the domains of MPAs and EDAs, and proposing a number of ways to combine them, we expect to contribute to an increasing trend in the application of MPAs–EDAs in particular and EAs in general.

References

Abbeel P, Koller D, Ng AY (2006) Learning factor graphs in polynomial time and sample complexity. J Mach Learn Res 7:1743–1788

Baluja S, Davies S (1997) Using optimal dependency-trees for combinatorial optimization: learning the structure of the search space. In: Fisher DH (ed) Proceedings of the 14th international conference on machine learning. Morgan Kaufmann, San Francisco, pp 30–38

Batra D, Gallagher A, Parikh D, Chen T (2010) Beyond trees: MRF inference via outer-planar decomposition. In: 2010 IEEE conference on computer vision and pattern recognition (CVPR). IEEE, San Francisco, pp 2496–2503

Bickson D (2008) Gaussian belief propagation: theory and application. arXiv preprint arXiv:0811.2518. Accessed 18 Dec 2014

Braunstein A, Mézard M, Zecchina R (2005) Survey propagation: an algorithm for satisfiability. Random Struct Algorithms 27(2): 201–226

Braunstein A, Mézard M, Zecchina R (2006) Constraint satisfaction by survey propagation. In: Percus A, Istrate G, Moore C (eds) Computational complexity and statistical physics. Oxford University Press, Oxford, pp 107–124

Brownlee AE, McCall JA, Shakya SK, Zhang Q (2010) Structure learning and optimisation in a Markov network based estimation of distribution algorithm. In: Exploitation of linkage learning in evolutionary algorithms. Springer, Berlin, pp 45–69

Brownlee AEI (2009) Multivariate Markov networks for fitness modelling in an estimation of distribution algorithm. PhD Thesis, The Robert Gordon University, School of Computing, Aberdeen

Brownlee AEI, McCall J, Pelikan M (2012) Influence of selection on structure learning in Markov network EDAs: an empirical study.

MEDAL Report No. 2012006. Missouri Estimation of Distribution Algorithms Laboratory (MEDAL)

Brownlee AEI, McCall J, Zhang Q, Brown D (2008) Approaches to selection and their effect on fitness modelling in an estimation of distribution algorithm. In: Proceedings of the 2008 congress on evolutionary computation CEC-2008. IEEE Press, Hong Kong, pp 2621–2628

Ceberio J, Irurozki E, Mendiburu A, Lozano JA (2012) A review on estimation of distribution algorithms in permutation-based combinatorial optimization problems. Prog Artif Intell 1(1): 103–117

Ceberio J, Mendiburu A, Lozano JA (2013) The Plackett–Luce ranking model on permutation-based optimization problems. In: 2013 IEEE congress on evolutionary computation (CEC). IEEE, Cancún, pp 494–501

Chen B, Hu J (2010a) An adaptive niching EDA based on clustering analysis. In: 2010 IEEE congress on evolutionary computation (CEC). IEEE, Barcelona, pp 1–7

Chen B, Hu J (2010b) A novel clustering based niching EDA for protein folding. In: Proceedings of the world congress on nature and biologically inspired computing, 2009. NaBIC 2009. IEEE, Coimbatore, pp 748–753

Crick C, Pfeffer A (2003) Loopy belief propagation as a basis for communication in sensor networks. In: Proceedings of the 19th annual conference on uncertainty in artificial intelligence (UAI-2003). Morgan Kaufmann Publishers, San Francisco, pp 159–166

Deming WE, Stephan FF (1940) On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. Ann Math Stat 11(4):427–444

Dolev D, Bickson D, Johnson J (2009) Fixing convergence of Gaussian belief propagation. In: IEEE international symposium on information theory, 2009. ISIT 2009. IEEE, Seoul, pp 1674–1678

Domínguez E, Lage-Castellanos A, Mulet R, Ricci-Tersenghi F, Rizzo T (2011) Characterizing and improving generalized belief propagation algorithms on the 2D Edwards–Anderson model. J Stat Mech Theory Exp 2011(12):P12007

Dong W, Yao X (2008) NichingEDA: utilizing the diversity inside a population of EDAs for continuous optimization. In: Proceedings of the 2008 congress on evolutionary computation CEC-2008. IEEE Press, Hong Kong, pp 1260–1267

Duchi J, Tarlow D, Elidan G, Koller D (2007) Using combinatorial optimization within max-product belief propagation. In: Advances in neural information processing systems 19: proceedings of the 2006 conference, vol 19. The MIT Press, Cambridge, p 369

Echegoyen C (2012) Contributions to the analysis and understanding of estimation of distribution algorithms. PhD Thesis, Department of Computer Science and Artificial Intelligence, University of the Basque Country

Echegoyen C, Lozano JA, Santana R, Larrañaga P (2007) Exact Bayesian network learning in estimation of distribution algorithms. In: Proceedings of the 2007 congress on evolutionary computation CEC-2007. IEEE Press, Los Alamitos, pp 1051–1058. doi:10. 1109/CEC.2007.4424586

Echegoyen C, Santana R, Lozano JA, Larrañaga P (2008) The impact of probabilistic learning algorithms in EDAs based on Bayesian networks. In: Linkage in evolutionary computation, studies in computational intelligence. Springer, Berlin, pp 109–139. doi:10.1007/978-3-540-85068-7_6

Echegoyen C, Mendiburu A, Santana R, Lozano JA (2009) Analyzing the probability of the optimum in EDAs based on Bayesian networks. In: Proceedings of the 2009 congress on evolutionary computation CEC-2009. IEEE Press, Trondheim, pp 1652–1659. doi:10.1109/CEC.2009.4983140

Echegoyen C, Mendiburu A, Santana R, Lozano JA (2010a) Analyzing the k most probable solutions in EDAs based on Bayesian networks. In: Exploitation of linkage learning in evolutionary algorithms, evolutionary learning and optimization. Springer, pp 163–189. doi:10.1007/978-3-642-12834-9_8

Echegoyen C, Mendiburu A, Santana R, Lozano JA (2010b) Estimation of Bayesian networks algorithms in a class of complex networks. In: Proceedings of the 2010 congress on evolutionary computation CEC-2010. IEEE Press, Barcelona. doi:10.1109/CEC.2010.5586511

Echegoyen C, Zhang Q, Mendiburu A, Santana R, Lozano JA (2011) On the limits of effectiveness in estimation of distribution algorithms. In: Proceedings of the 2011 congress on evolutionary computation CEC-2007. IEEE Press, New Orleans, pp 1573–1580. doi:10.1109/CEC.2011.5949803

Echegoyen C, Mendiburu A, Santana R, Lozano JA (2012) Toward understanding EDAs based on Bayesian networks through a quantitative analysis. IEEE Trans Evol Comput 16(2):173–189. doi:10.1109/TEVC.2010.2102037

Etxeberria R, Larrañaga P (1999) Global optimization using Bayesian networks. In: Ochoa A, Soto MR, Santana R (eds) Proceedings of the second symposium on artificial intelligence (CIMAF-99). Editorial Academia, Havana, pp 332–339

Frey BJ, Dueck D (2007) Clustering by passing messages between data points. Science 315:972–976

Furtlehner C, Schoenauer M (2010) Multi-objective 3-SAT with survey-propagation. In: Proceedings of the NIPS 2010 workshop on discrete optimization in machine learning: structures, algorithms and applications (DISCML), Whistler, Canada. http://hal.inria.fr/inria-00533149 Accessed 18 Dec 2014

Gao Y, Culberson JC (2005) Space complexity of estimation of distribution algorithms. Evol Comput 13(1):125–143

Givoni IE, Frey BJ (2009) A binary variable model for affinity propagation. Neural Comput 21(6):1589–1600

Givoni IE, Chung C, Frey BJ (2011) Hierarchical affinity propagation. In: Proceedings of the 27th annual conference on uncertainty in artificial intelligence (UAI-2011). Morgan Kaufmann, Barcelona

Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley, Reading

González C, Lozano JA, Larrañaga P (2002) Mathematical modeling of UMDAc algorithm with tournament selection. Behaviour on linear and quadratic functions. Int J Approx Reason 31(4):313–340

Grahl J, Minner S, Bosman PA (2008) Learning structure illuminates black boxes—an introduction to estimation of distribution algorithms. In: Advances in metaheuristics for hard optimization. Springer, Berlin, pp 365–395

Harik G (1999) Linkage learning via probabilistic modeling in the ECGA. IlliGAL Report 99010. University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana

Hartmann AK, Weigt M (2005) Phase transitions in combinatorial optimization problems: basics, algorithms and statistical mechanics. Wiley, Weinheim

Helmi BH, Rahmani AT, Pelikan M (2014) A factor graph based genetic algorithm. Int J Appl Math Comput Sci 24(3):621–633

Henrion M (1988) Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In: Lemmer JF, Kanal LN (eds) Proceedings of the second annual conference on uncertainty in artificial intelligence. Elsevier, Amsterdam, pp 149–164

Heskes T (2004) On the uniqueness of belief propagation fixed points. Neural Comput 16:2379–2413

Holland JH (1975) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. University of Michigan Press, Ann Arbor

Höns R (2006) Estimation of distribution algorithms and minimum relative entropy. PhD Thesis, University of Bonn, Bonn

Höns R (2012) Using maximum entropy and generalized belief propagation in estimation of distribution algorithms. In: Shakya S, Santana R (eds) Markov networks in evolutionary computation. Springer, Berlin, pp 175–190

Höns R, Santana R, Larrañaga P, Lozano JA (2007) Optimization by max-propagation using Kikuchi approximations. Technical Report EHU-KZAA-IK-2/07. Department of Computer Science and Artificial Intelligence, University of the Basque Country

Ihler AT, Fisher J, Willsky AS (2006) Loopy belief propagation: convergence and effects of message errors. J Mach Learn Res 6(1):905

Jiroušek R, Přeučil S (1995) On the effective implementation of the iterative proportional fitting procedure. Comput Stat Data Anal 19:177–189

Johnson A, Shapiro JL (2002) The importance of selection mechanisms in distribution estimation algorithms. In: Collet P (ed) Proceedings of EA 2001, lecture notes in computer science, vol 2310. Springer, pp 91–103

Kaban A, Bootkrajang J, Durrant RJ (2013) Towards large scale continuous EDA: a random matrix theory perspective. In: Proceedings of the genetic and evolutionary computation conference GECCO-2013. ACM, Amsterdam, pp 383–390

Karshenas H, Santana R, Bielza C, Larrañaga P (2011) Multi-objective optimization with joint probabilistic modeling of objectives and variables. In: Evolutionary multi-criterion optimization: sixth international conference, EMO 2011, lecture notes in computer science. Springer, Berlin, pp 298–312. doi:10.1007/978-3-642-19893-9_21

Karshenas H, Santana R, Bielza C, Larrañaga P (2012) Continuous estimation of distribution algorithms based on factorized Gaussian Markov networks. In: Shakya S, Santana R (eds) Markov networks in evolutionary computation. Springer, Berlin, pp 157–173. doi:10.1007/978-3-642-28900-2-10

Kim K, McKay BR, Punithan D (2010) Sampling bias in estimation of distribution algorithms for genetic programming using prototype trees. In: PRICAI 2010: trends in artificial intelligence. Springer, Berlin, pp 100–111

Kroc L, Sabharwal A, Selman B (2009) Message-passing and local heuristics as decimation strategies for satisfiability. In: Proceedings of the 2009 ACM symposium on applied computing. ACM, Honolulu, pp 1408–1414

Kschischang FR, Frey BJ, Loeliger HA (2001) Factor graphs and the sum–product algorithm. IEEE Trans Inf Theory 47(2):498–519

Larrañaga P, Lozano JA (eds) (2002) Estimation of distribution algorithms. A new tool for evolutionary computation. Kluwer Academic Publishers, Boston

Leone M, Weigt S, Weigt M (2007) Clustering by soft-constraint affinity propagation: applications to gene-expression data. Bioinformatics 23(20):2708–2715

Lima CF (2009) Substructural local search in discrete estimation of distribution algorithms. PhD Thesis, University of Algarve

Lima CF, Pelikan M, Goldberg DE, Lobo FG, Sastry K, Hauschild M (2007) Influence of selection and replacement strategies on linkage learning in BOA. In: Proceedings of the 2007 congress on evolutionary computation CEC-2007. IEEE Press, Los Alamitos, pp 1083–1090

Lima CF, Pelikan M, Lobo FG, Goldberg DE (2009) Loopy substructural local search for the Bayesian optimization algorithm. In: Engineering stochastic local search algorithms. Designing, implementing and analyzing effective heuristics. Springer, Berlin, pp 61–75

Lozano JA, Larrañaga P, Inza I, Bengoetxea E (eds) (2006) Towards a new evolutionary computation: advances on estimation of distribution algorithms. Springer, Heidelberg

Mahfoud SW (1995) Niching methods for genetic algorithms. PhD Thesis, University of Illinois at Urbana-Champaign, Urbana. Also IlliGAL Report No. 95001

Malioutov DM, Johnson JK, Willsky AS (2006) Walk-sums and belief propagation in Gaussian graphical models. J Mach Learn Res 7:2031–2064

Meltzer T, Yanover C, Weiss Y (2005) Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation. In: Tenth IEEE international conference on computer vision, pp 428–435

Mendiburu A, Lozano J, Miguel-Alonso J (2005) Parallel implementation of EDAs based on probabilistic graphical models. IEEE Trans Evol Comput 9(4):406–423

Mendiburu A, Santana R, Bengoetxea E, Lozano, J (2007a) A parallel framework for loopy belief propagation. In: Thierens D et al (eds) Proceedings of the genetic and evolutionary computation conference GECCO-2007, vol II. Companion material. ACM Press, London, pp 2843–2850. http://dl.acm.org/citation.cfm?id=1274084 Accessed 18 Dec 2014

Mendiburu A, Santana R, Lozano JA (2007b) Introducing belief propagation in estimation of distribution algorithms: a parallel framework. Technical Report EHU-KAT-IK-11/07. Department of Computer Science and Artificial Intelligence, University of the Basque Country

Mendiburu A, Santana R, Lozano JA (2012) Fast fitness improvements in estimation of distribution algorithms using belief propagation. In: Santana R, Shakya S (eds) Markov networks in evolutionary computation. Springer, Berlin, pp 141–155. doi:10.1007/978-3-642-28900-2-9

Mézard M, Parisi G, Zechina R (2002) Analytic and algorithmic solution of random satisfiability problems. Science 297:812–812. doi:10.1126/science.1073287

Minka T (2001) A family of algorithms for approximate bayesian inference. PhD Thesis, Massachusetts Institute of Technology

Minka T (2005) Divergence measures and message passing. Technical Report TR-2005-173. Mitsubishi Electric Research Laboratories

Mooij JM (2005) Validity estimates for loopy belief propagation on binary real-world networks. In: Advances in neural information processing systems 17. MIT Press, Cambridge, pp 945–952

Mooij J (2010) libDAI: a free and open source C++ library for discrete approximate inference in graphical models. J Mach Learn Res 11:2169–2173

Mühlenbein H (2012) Convergence theorems of estimation of distribution algorithms. In: Shakya S, Santana R (eds) Markov networks in evolutionary computation. Springer, Berlin, pp 91–108

Mühlenbein H, Höns R (2006) The factorized distributions and the minimum relative entropy principle. In: Pelikan M, Sastry K, Cantú-Paz E (eds) Scalable optimization via probabilistic modeling: from algorithms to applications, studies in computational intelligence. Springer, Berlin, pp 11–38

Mühlenbein H, Mahnig T (2002) Evolutionary optimization and the estimation of search distributions with applications to graph bipartitioning. Int J Approx Reason 31(3):157–192

Mühlenbein H, Paaß G (1996) From recombination of genes to the estimation of distributions I. Binary parameters. In: Voigt HM, Ebeling W, Rechenberg I, Schwefel HP (eds) Parallel problem solving from nature—PPSN IV, lectures notes in computer science, vol 1141. Springer, Berlin, pp 178–187

Mühlenbein H, Mahnig T, Ochoa A (1999) Schemata, distributions and graphical models in evolutionary optimization. J Heuristics 5(2):213–247

Munetomo M, Murao N, Akama K (2008) Introducing assignment functions to Bayesian optimization algorithms. Inf Sci 178(1):152–163

Murphy KP, Weiss Y, Jordan MI (1999) Loopy belief propagation for approximate inference: an empirical study. In: Proceedings of the fifteenth conference on Uncertainty in artificial intelligence. Morgan Kaufmann Publishers, Inc., San Francisco, pp 467–475

Nilsson D (1998) An efficient algorithm for finding the M most probable configurations in probabilistic expert systems. Stat Comput 2:159–173

Ocenasek J, Schwarz J (2002) Estimation of distribution algorithm for mixed continuous–discrete optimization problems. In: Proceedings of the 2nd Euro-international symposium on computational intelligence. IOS Press, Kosice, pp 227–232

Ochoa A, Höns R, Soto MR, Mühlenbein H (2003) A maximum entropy approach to sampling in EDA—the single connected case. In: Progress in pattern recognition, speech and image analysis, lectures notes in computer science, vol 2905. Springer, Berlin, pp 683–690

Pearl J (1988) Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann, San Mateo

Pelikan M, Goldberg DE, Lobo F (2002) A survey of optimization by building and using probabilistic models. Comput Optim Appl 21(1):5–20

Peña J, Lozano JA, Larrañaga P (2005) Globally multimodal problem optimization via an estimation of distribution algorithm based on unsupervised learning of Bayesian networks. Evol Comput 13(1):43–66

Ponce de León E, Díaz E (2012) Adaptive evolutionary algorithm based on a cliqued gibbs sampling over graphical Markov model structure. In: Shakya S, Santana R (eds) Markov networks in evolutionary computation. Springer, Berlin, pp 109–123

Regnier-Coudert O (2013) Bayesian network structure learning using characteristic properties of permutation representations with applications to prostate cancer treatment. PhD Thesis, Robert Gordon University

Rivera JP, Santana R (2000) Design of an algorithm based on the estimation of distributions to generate new rules in the XCS classifier system. Technical Report ICIMAF 2000-100, CEMA-FIT 2000-78. Institute of Cybernetics, Mathematics and Physics, Havana

Santana R (2003) Factorized Distribution Algorithms: selection without selected population. Technical Report ICIMAF 2003-240. Institute of Cybernetics, Mathematics and Physics, Havana

Santana R (2006) Advances in probabilistic graphical models for optimization and learning. Applications in protein modelling. PhD Thesis, University of the Basque Country

Santana R, Larrañaga P, Lozano JA (2005) Interactions and dependencies in estimation of distribution algorithms. In: Proceedings of the 2005 congress on evolutionary computation CEC-2005. IEEE Press, Edinburgh, pp 1418–1425. doi:10.1109/CEC.2005.1554856

Santana R, Mendiburu A, Lozano JA (2008) An empirical analysis of loopy belief propagation in three topologies: grids, small-world networks and random graphs. In: Jaeger M, Nielsen TD (eds) Proceedings of the fourth European workshop on probabilistic graphical models (PGM-2008), pp 249–256

Santana R, Larrañaga P, Lozano JA (2010) Learning factorizations in estimation of distribution algorithms using affinity propagation. Evol Comput 18(4):515–546. http://www.mitpressjournals.org/doi/abs/10.1162/EVCO_a_00002. Accessed 18 Dec 2014

Santana R, Mendiburu A, Lozano JA (2012a) Evolving NK-complexity for evolutionary solvers. In: Companion proceedings of the 2012 genetic and evolutionary computation conference GECCO-2012. ACM Press, Philadelphia, pp 1473–1474. http://dl.acm.org/citation.cfm?id=2330997. Accessed 18 Dec 2014

Santana R, Mendiburu A, Lozano JA (2012b) New methods for generating populations in Markov network based EDAs: decimation strategies and model-based template recombination. Technical Report EHU-KZAA-TR:2012-05. Department of Computer Science and Artificial Intelligence, University of the

Basque Country. http://hdl.handle.net/10810/9180. Accessed 18 Dec 2014

Santana R, Mendiburu A, Lozano JA (2013) Message passing methods for estimation of distribution algorithms based on Markov networks. In: Proceedings of the 4th conference on swarm, evolutionary, and memetic computing (SEMCCO-2013), lectures notes in computer science. Springer, Chennai, pp 419–430 (in press)

Sastry K, Abbass HA, Goldberg DE, Johnson D (2005) Sub-structural niching in estimation of distribution algorithms. In: Proceedings of the 2005 conference on genetic and evolutionary computation. ACM, Washington, DC, pp 671–678

Sastry K, Lima CF, Goldberg DE (2006) Evaluation relaxation using substructural information and linear estimation. In: Proceedings of the 8th annual conference on genetic and evolutionary computation. ACM, New York, pp 419–426

Sastry K, Goldberg DE, Llorá X (2007) Towards billion-bit optimization via a parallel estimation of distribution algorithm. In: Thierens D et al (eds) Proceedings of the genetic and evolutionary computation conference GECCO-2007, vol I. ACM Press, London, pp 577–584

Sato H, Hasegawa Y, Bollegala D, Iba H (2012) Probabilistic model building GP with belief propagation. In: 2012 IEEE congress on evolutionary computation (CEC). IEEE, pp 1–8

Shakya S, Santana R, Lozano JA (2012) A Markovianity based optimisation algorithm. Genet Program Evol Mach 13(2):159–195. doi:10.1007/s10710-011-9149-y

Soto MR (2003) A single connected factorized distribution algorithm and its cost of evaluation. PhD Thesis, University of Havana, Havana (in Spanish)

Suwannik W, Chongstitvatana P (2008) Solving one-billion-bit noisy OneMax problem using estimation distribution algorithm with arithmetic coding. In: Proceedings of the 2008 congress on evolutionary computation CEC-2008. IEEE Press, Hong Kong, pp 1203–1206

Tanaka K, Shouno H, Okada M, Titterington D (2004) Accuracy of the Bethe approximation for hyperparameter estimation in probabilistic image processing. J Phys A 37(36):8675

Valdez-Peña IS, Hernández-Aguirre A, Botello-Rionda S (2009) Approximating the search distribution to the selection distribution in EDAs. In: Proceedings of the genetic and evolutionary computation conference GECCO-2009. ACM, New York, pp 461–468

Van Hoyweghen C, Goldberg D, Naudts B (2002a) From twomax to the Ising model: easy and hard symmetrical problems. In: Proceedings of the genetic and evolutionary computation conference GECCO-2002. Morgan Kaufmann Publishers, Inc., San Francisco, pp 626–633

Van Hoyweghen C, Naudts B, Goldberg D (2002b) Spin-flip symmetry and synchronization. Evol Comput 10(4):317–344

Wainwright MJ, Jordan MI (2003) Graphical models, exponential families, and variational inference. Technical Report 649. Department of Statistics, University of California, Berkeley

Wainwright M, Jaakkola T, Willsky A (2002) Exact MAP estimates by (hyper) tree agreement. Adv Neural Inf Process Syst 15:809–816

Wainwright M, Jaakkola T, Willsky A (2004) Tree consistency and bounds on the performance of the max-product algorithm and its generalizations. Stat Comput 14:143–166

Wang Z, Zoghi M, Hutter F, Matheson D, De Freitas N (2013) Bayesian optimization in high dimensions via random embeddings. In: Proceedings of the Twenty-Third international joint conference on Artificial Intelligence. AAAI Press, Chicago, pp 1778–1784

Weiss Y (2000) Correctness of local probability propagation in graphical models with loops. Neural Comput 12:1–41

Weiss Y, Freeman WT (2001) On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. IEEE Trans Inf Theory 47(2):723–735

Welling M (2004) On the choice of regions for generalized belief propagation. In: Proceedings of the 20th conference on uncertainty in artificial intelligence (UAI-2004). Morgan Kaufmann Publishers, Banff, pp 585–592

Wiegerinck W, Heskes T (2003) Fractional belief propagation. In: Advances in neural information processing systems. MIT, Vancouver, pp 455–462

Xing EP, Jordan MI (2003) Graph partition strategies for generalized mean field inference. Technical Report CSD-03-1274. Division of Computer Science, University of California, Berkeley

Yanover C, Weiss Y (2003) Approximate inference and protein-folding. In: Becker S, Thrun S, Obermayer K (eds) Advances in neural information processing systems 15. MIT Press, Cambridge, pp 1457–1464

Yedidia JS, Freeman WT, Weiss Y (2005) Constructing free energy approximations and generalized belief propagation algorithms. IEEE Trans Inf Theory 51(7):2282–2312

Yuille A (2001) A double-loop algorithm to minimize the Bethe and Kikuchi free energies. Neural Comput 14(6):1691–1722

Zilberstein S (1996) Using anytime algorithms in intelligent systems. AI Mag 17(3):73