We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

Our authors are among the

154
Countries delivered to

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



12

Solving Combinatorial Problems with Time Constrains using Estimation of Distribution Algorithms and Their Application in Video-Tracking Systems

Antonio Berlanga, Miguel A. Patricio, Jesús García, and José M. Molina
Applied Artificial Intelligence Group. Universidad Carlos III de Madrid
Colmenarejo, Madrid,
Spain

1. Introduction

EDAs (Estimation of Distribution Algorithms) present the suitable features to deal with problems requiring a very efficient search: small populations and a few iterations, compared with the more classic approaches to Evolutionary Algorithms (EAs). The fundamental difference of EDAs with classical EAs is that the formers carry out a search of the probability distribution describing the optimal solutions while EAs directly make the search and provide the solutions to the problem with the solutions itself. They share the necessity of codification of solutions by means of binary chains, in the EA terminology they are the "individuals" and the definition of a merit measurement that allows to orient the search direction, the so called "fitness function". In the case of EDAs, operators to manipulate individuals in the search, such as mutation, selections, and crossover, are not needed, since the search is performed directly on the distribution which describes all possible individuals.
In this chapter, authors will evaluate the efficiency of EDAs to solve combinatorial problems with time constrains. Specifically, authors will model the visual data association for real-time video tracking problem as a combinatorial problem. Before the application of EDAs to this real-life combinatorial problem, the authors will discuss the application of EDAs algorithms to a classical combinatorial problem, such as the 0/1 knapsack problem, in order to know the complexity degree of the association problem and to find out the most suitable parameters for real-time video tracking problem [1].
The outline of the chapter will be as follows. First, several EDA algorithms will be presented and their evaluation using the theoretical combinatorial problem of 0/1 knapsack problem, which has similar complexity to the association problem in video tracking systems. Next, the mathematical formulation of the Data Association Problem will be shown. Then, the applications of EDA to data association problem, defining the heuristic and the codification, will be presented. Finally, the authors will show the experiments compare the behaviour of several algorithms, taking the advanced Particles-MS tracking as benchmark, in three scenarios taken from two different sources: the publicly available CVBASE [2] and a DV camcorder.

Source: New Achievements in Evolutionary Computation, Book edited by: Peter Korosec,
ISBN 978-953-307-053-7, pp. 318, February 2010, INTECH, Croatia, downloaded from SCIYO.COM

2. Estimation of Distributions Algorithms (EDAs)

The Estimation of Distribution Algorithms (EDAs) [3] are a family of evolutionary algorithms which represents an alternative to the classical optimization methods. Algorithmically, a Genetic Algorithm and an EDA only differ in the procedure to generate new individuals. EDAs replace the use of an evolving population by a vector that directly codifies the joint probability distribution of vectors corresponding to the best solutions. The crossover and mutation operators are replaced by rules that update the probability distribution. A great advantage of the EDAs on the evolutionary algorithms is that they allow expressing the interactions between variables of the problem by means of the associated joint probability distribution. In addition, they improve the time of convergence and the necessary space of memory for its operation. The algorithm of an EDA is sketched in the following graph.

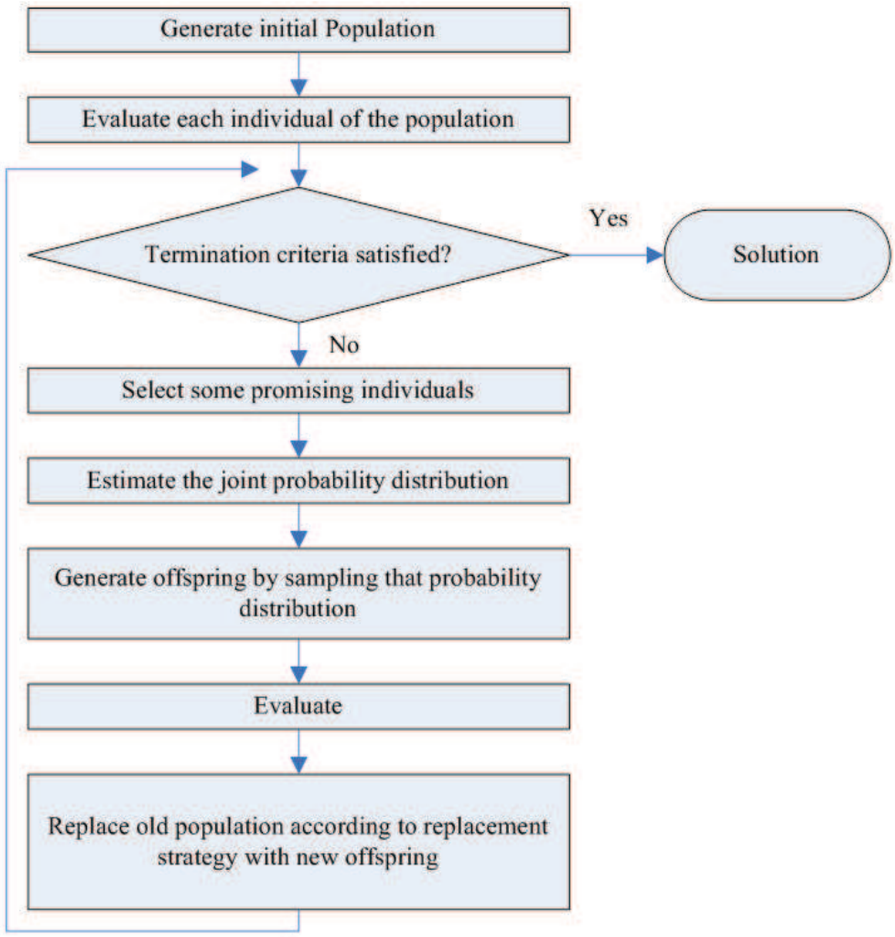


Fig. 1. High level outline of EDA steps

The key point of the use of EDAs is in the estimation of the joint probability distribution. The simplest situation is that in which the joint probability distribution factorizes as a product of univariate and independent distributions, that is to say, there is no dependency between the variables. In this situation the estimation of the probability distribution is made using the marginal frequencies. The problem of association exposed in this work allows this

treatment and bases the use of EDAs by the characteristic that allows solving an optimization problem quickly [4], [5]. In the next section, we describe the EDAs more studied and applied to univariate problems and in the appendix the code of each EDA is detailed.

2.1 Description of EDAs algorithms

In this section, we depict the several concepts on the algorithm used in our work. These algorithms are UMDA (Univariate Marginal Distribution Algorithm) [6], PBil (Population-Based Incremental Learning) [7] and cGA (The Compact Genetic Algorithm) [8].

2.1.1 UMDA. Univariate Marginal Distribution Algorithm

In the UMDA [6] the joint probability distribution is estimated as the relative frequencies of the variables' values stored in a data set. Independence between the variables is assumed and theoretically it has been demonstrated that UMDA works almost perfectly with linear problems and rather well when the dependencies are little significant.

In UMDA can appear some problems associated to the genetic drift, and some modifications have been proposed such as the correction of Laplace to the calculation of the relative frequency [9]. In this case the relative frequency is estimated as:

$$p_l(x) = \prod_{i=1}^{n} \frac{\sum_{j=1}^{N} \delta_j\left(X_i = x_i \mid D_{l-1}^{Se}\right) + 1}{N + r_i}, \qquad r_i \text{ different values of } X_i \tag{1}$$

There are theoretical evidences that demonstrate that UMDA approximates their behaviour to a canonical genetic algorithm (GA) with uniform crossover.

2.1.2 PBIL. Population-Based Incremental Learning

The PBIL (Population-Based Incremental Learning) [7] mixes the search applied by genetic algorithms with the competitive learning, applying a Hebbian rule to update the vector of probabilities. It has been empirically demonstrated that PBil works equal or better than GA in problems in which GA works well and fails in the problems that GA fails. The main difference with GA is that PBIL does not use a population of solutions that evolves, replacing it by a vector of probabilities. This change presents some advantages:

- The algorithm is simpler, for example it does not require arrangements.
- The capacity of representation by means of a vector of probabilities in PBIL is minor who the one of a population of solutions in GA, therefore the convergence is faster. Search in exploration is sacrificed to have a higher rate of convergence.

In order to apply the PBil algorithm it is necessary to give value to four parameters: population size, learning rate, probability of mutation and mutation rate. To the association problem the probability of mutation is set to zero with the purpose of accelerating the convergence of the algorithm.

The main parameter that will affect the speed of convergence is the learning rate, whichever greater is its value, more quickly finalizes the search, losing, of course, quality in the solutions. The PBil algorithm originally was oriented to functions optimization, but has been efficient treating combinatorial optimization problems of complexity NP, in terms to find solutions of the same quality that GA needing smaller number of function evaluations.

2.1.3 CGA. The Compact Genetic Algorithm

The cGA [8] simulates the performance of a canonical genetic algorithm with uniform crossover. It is the simplest EDA, and only needs a parameter, the population size that has the effect of a learning rate.

As the authors of CGA suggest, this algorithm has an important additional utility, allows discriminating quickly when a problem is easy to be treated by means of evolutionary algorithms.

3. Adjustment of the EDA parameters using KP0/1 problem

In order to fit the parameters of each one of EDAs applied to tracking systems, we evaluated them using a theoretical combinatorial problem with similar complexity and well-known optimal solution. The objective for these theoretical experiments is two-fold:

- Compare the number of function evaluations that needs each algorithm to obtain solutions of a given quality. This efficiency metric allows us choose the suitable algorithm for the real-time video association problem
- Fit the parameters of each algorithm to obtain the best relation between the quality of solutions and speed of execution

It has been taken three KP0/1 problems [10] with increasing difficulty (10, 20 and 40 objects). These sizes correspond with the practical problem dealt since the codification of the association matrix will be, in the cases of higher-density also around 40. The size of the search space scales with 2^n. For the three problems, the optimal solution is known and not exist correlation between the variables; this is the worst case. The knapsack 0/1 problem of (KP0/1) is a NP-complete combinatorial optimization problem and, for example, the difficulty to be solved was used to make the first algorithm of generalized public key encryption [11]. The knapsack 0/1 problem is defined as:

$$\begin{aligned} &\text{Maximize} \sum_{j=1}^{n} c_j x_j \\ &\text{Subject to} \sum_{j=1}^{n} a_j x_j \le b \\ &x_j \in \{0,1\}, \forall j, c_j > 0, a_j > 0, \sum_j c_j > b, \max_j \{a_j\} \le b \end{aligned} \tag{2}$$

The following figures contain the comparison of performances by UMDA, PBIL, CGA and the canonical GA with uniform applied in this work. The number of evaluations of fitness necessary to obtain a certain probability of success has been plotted. An algorithm is successful when it finds the optimal solution and the success probability is calculated with the frequency of success in 50 executions with different random seeds. The parameters of the different algorithms are obtained by trail-and-error treating to diminish the number of fitness evaluations.

The graphs show, for all the algorithms and problems, the exponential growth of the number of evaluations when it is wanted to reach great percentage of success. In the three KP0/1 the UMDA has obtained better results, it needs fewer evaluations than the rest to obtain solutions of the same quality. The PBIL has a behaviour similar to the UMDA treating

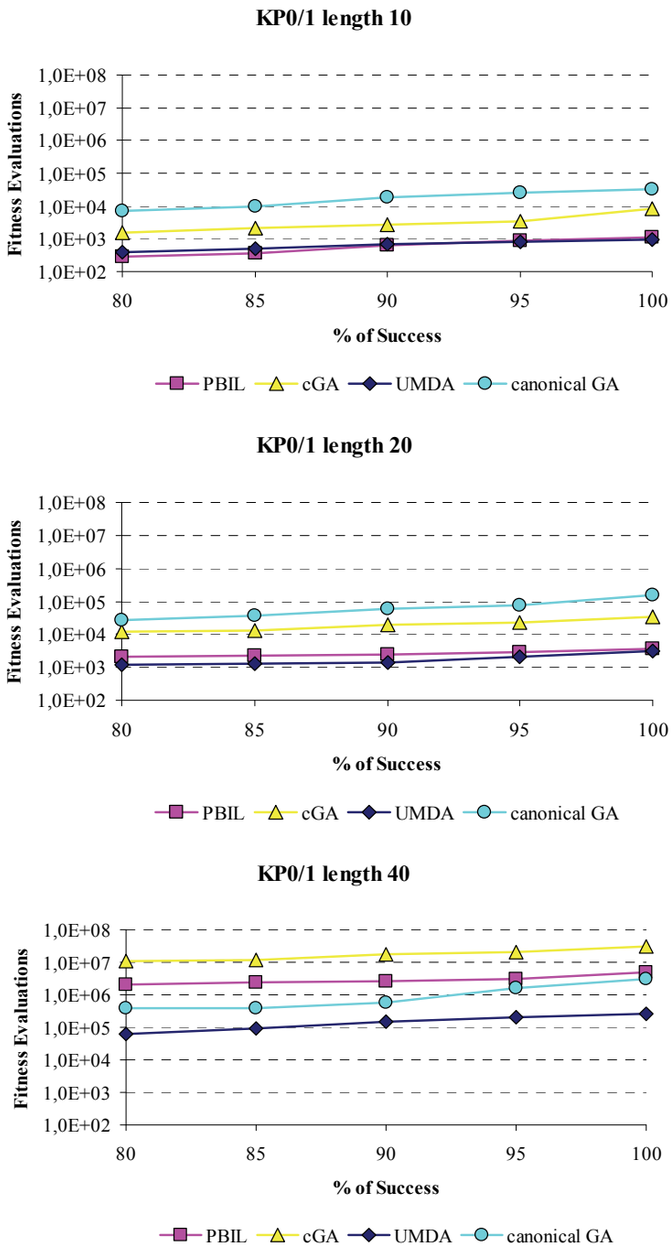


Fig. 2. Comparison of the number of fitness evaluations versus the probability of success of the EDAS and canonical GA to solve several KP0/1 of different complexity

the simplest problems but it behaves worse than the canonical GA when it is applied to the most difficult problem. The CGA shows the worse behaviour, if for the two simpler problems has an intermediate behaviour between UMDA and the canonical GA, for the KP0/1 of 40 objects has instability respect to the parameter that represents the size of the population, N.

In the following figure, the variation of the number of evaluations of fitness based on the difficulty of the problem when the percentage of success is the 100% has been plotted.

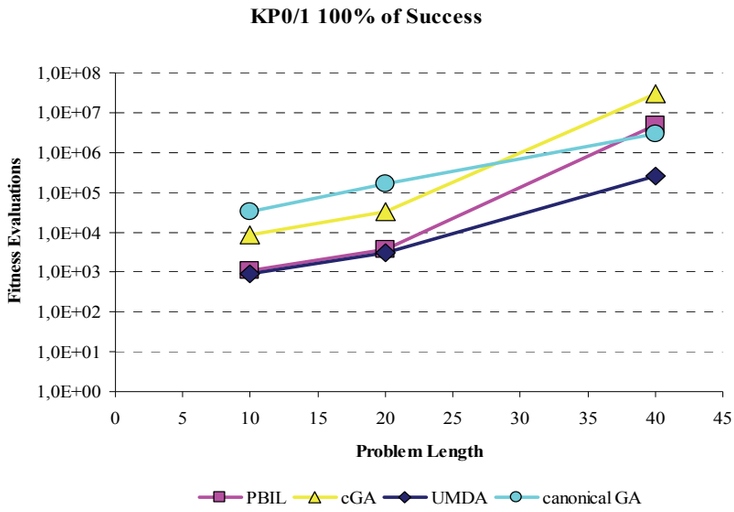


Fig. 3. Evolution of the number of evaluations of fitness based on the length of the problem.

It is observed that the UMDA is the algorithm with smaller number of evaluations of fitness needs to reach the 100% of success. The behaviour of the canonical GA is remarkable, it scales an order of magnitude whenever the difficulty of the problem is duplicated, while this increase is greater for the rest of algorithms. However, for simpler problems the canonical GA needs almost two orders of magnitude more evaluations of fitness to obtain the same quality.

As it has been previously indicated, the objective of this work is to solve the combinatorial optimization problem that appears in the association problem from blobs to tracks for real-time video tracking. The analysis of the EDAs and canonical GA on KP0/1 has allowed us to discriminate the algorithm that presents greater potential of application and its parameters. These parameters are in Table 1 and will be used later to solve the association problem. In section 5, the results of these algorithms applied to the association problem will be presented.

	Parameters that were applied in KP0/1		
	Length 10	**Length 20**	**Length 40**
PBIL	N=1 M=9 LR=0.05	N=1 M=20 LR=0.05	N=8 M=1000 LR=0.05
CGA	N=50	N=200	N=20000
UMDA	N=1 M=9	N=8 M=50	N=8 M=250
GA	M=100 $P_{mutation}$=0.01 $P_{crossover}$=0.9 Elitism=20%	M=200 $P_{mutation}$=0.01 $P_{crossover}$=0.9 Elitism=20%	M=800 $P_{mutation}$=0.01 $P_{crossover}$=0.9 Elitism=20%

Table 1. Values of the parameters of the EDAs and GA that allow to obtain a 100% of success in the problem of the KP0/1

4. Formulation of data association problem

The data association problem, also known as motion correspondence in the case of video analysis, is the most important challenge in multi-target video tracking systems. The complexity of scenarios, expressed in the number of interacting targets, irregular motion, presence of background noise, etc., presents a collection of problems that must be solved to guarantee a robust performance and so reliable output with a certain level of accuracy and stability.

Among alternative approaches, Bayesian inference methods have gained a high reputation for visual processing and estimation [12], [13], [14], due the potential to provide a solution as close to the theoretically optimal as possible. Closeness to theoretical optimum will depend on the computation capability to execute numeric approximations and also on the reliability of statistical models for target appearance, dynamics, likelihoods, a priori densities, etc. It usually becomes an intractable approach for most realistic situations and it is necessary the use of heuristic approaches and simplifications.

4.1 Problem statement. Motion correspondence as a search

The visual tracking problem involves an environment that changes with time [15]. The estimation of the number of objects in a scene, together with their instantaneous location, cinematic state and additional characteristics (shape, colour, identification, etc.) is the problem addressed by a visual tracker. The objects in the scene can be defined as a set of entities described by a set of characteristics that evolve in time (position, velocity, colour, shape, temperature, etc.). In this sense, environment could be defined in an instant as a set of objects, where each object is defined by a set of characteristics in this instant:

$$E[k]= \{O_1[t],\ldots, O_N[t]\} \tag{3}$$

So there are N[k] real objects moving in the covered area at time instant t[k].

The description of the objects is expressed in a vector state space, $\bar{x}_i \in \Re^d$. For instance a common simplified representation of objects in 2D camera plane contains position of object centroid, together with bounds (width and length) and their velocity and scale derivatives: $\bar{x}_i = [x \quad y \quad vx \quad vy \quad w \quad h \quad vw \quad vh]^t$ (d=8).

In the case of visual sensor, the phase of image preprocessing acquires some characteristics of the objects (observation), perturbed by the measurement process. The acquisition process is related with the digital image acquisition and the detection process. The first one defines the resolution in pixels and frames by second of the sequence of captured images; the second one defines the objects in the captured image. In the case of visual sensor, observed characteristic are more complex than conventional sensors, for example: color (or gray level), the shape, the skeleton, the contour, etc. and the position is an estimation from the detected object (typically the centroid). We will consider in this work as preprocessing phase the background subtraction to detect moving objects in monocular images [12]. After background subtraction and thresholding, we have a binary image where a detected object is observed through a set of compact regions (blobs), where a blob is a set of adjacent binary detected pixels in this instant:

$$Z^i[k] = \{b_1^i[k],\ldots, b_{M^i}^i[k]\} \tag{4}$$

where M^i is the number of blobs that are due to i-th object (unobservable).
The problem is that both N[k] and i superindex in observations are hidden, they must be deduced from the data. The only observable amount is the global set of blobs appearing in the foreground binary image: $Z[k]=\{b_1[k],\ldots b_M[k]\}$. So, the basic problem in video data association is the re-connection of blobs to be used to update tracks, searching the collection of blobs corresponding to each track $\bar{x}_i[k]$, $Z^i[k]$. The final object characteristics result from the combination of the blob characteristics that belong to the object
As mentioned in the introduction, target splitting and merging is a distinguishing problem in video data association with respect to other sensors. With other sensors such as radar, each acquired data, a plot, comes from a single object but, in visual sensor, acquired data, a blob, could be the result of several objects (merging), and several blobs can be originated by the same object (split). Blobs corresponding to separate objects can be merged when they come close together in the image, splitting again when they separate. The problem gets more complex since the image may contain false observations due to noise or target fragmentation, so that the total observations may be some arbitrarily large set of observations. This makes it difficult to provide unique track labels for different interacting objects, which is a fundamental capacity required for the usual applications of machine vision such as surveillance, scene analysis (sports), automatic annotation, etc. The problem would be aminorated by using a camera installed in a pole high enough so objects are viewed from near vertical geometry but this is often not possible in practical surveillance configurations (for instance, indoor scenes)
A Bayesian framework to determine the best estimation, X[k], inferred from available measurements, Z[k], is the one targeted at obtaining the maximum a posteriori probability of estimated state, conditioned to the whole set of observations:

$$\hat{X}[k] = \arg\max_{X[k]} P(X[k] \mid Z[k], Z[k-1], \ldots, Z[0]) \tag{5}$$

Where $\hat{X}[k]$ denotes both the number of targets and their state in the scene at time instant t[k],
$\hat{X}[k] = \hat{x}_{1\ldots N_k}[k] = \{\hat{x}_1[k], \ldots, \hat{x}_{N_k}[k]\}$, where $\hat{x}^i[k] \in \Re^d$, in our case d=8 as indicated above. Notice that objects forming state X[k] have a set structure instead of array, since the order in X[k]to represent the real objects is irrelevant.
So a joint estimation problem appears about the number of objects, N[k], and their parameters, $\hat{x}_i[k]$. A complete problem formulation for the multi-target joint estimation problem would take into account the different configurations in the number of objects and their characteristics (shape and motion state). The classical inference formulation applies the Bayes theorem to rearrange the problem in a recursive formulation:

$$\begin{aligned} &P(X[k] \mid Z[k], Z[k-1], \ldots, Z[0]) = \\ &\frac{1}{c} P(Z[k] \mid X[k]) \int \left[P(X[k] \mid X[k-1]) P(X[k-1] \mid Z[k-1], \ldots, Z[0]) \right] dX[k-1] \end{aligned} \tag{6}$$

where the integral in the joint problem would extend over the whole space of predicted state, P(X[k]|X[k-1]) considering both the number of objects and their corresponding states, and c is the normalization constant to guarantee the result is a probability density. In this formulation, and dropping time index for simplicity, P(Z|X) is the probability of observing a particular image Z given a certain current state X. It is the likelihood function and has a fundamental impact in the result. In our case we will particularize the observation process to the analysis of the binarized image resulting from the background subtraction and thresholding, so that

$$Z[k]=\{b_1,\ldots,b_{Mk}[k]\} \tag{7}$$

This multiple-target tracking problem can be split into two interdependent problems, data association (or motion correspondence) between measurements and objects, and state estimation, to update vectors $\hat{x}_j[k],\ j\in\{1,\ldots,N[k]\}$ with the assigned measurements.

Data association is the sequential decision process which takes, for each frame, the available measurements and assigns to the tracked targets up to that time instant. The assignment problem can be considered as part of the maximization of a posteriori likelihood of observations. So, if we consider a particular configuration of X[k] with N[k] tracks, its likelihood will depend on the sequential series of data assignments:

$$P(Z[k]|X[k]) = P(Z[k]|A[k],Z[k-1],A[k-1],Z[k-2],...,A[0],Z[0]) \tag{8}$$

where the assignment matrix $A[k]=\{a_{ij}[k]\}$ is defined as $a_{ij}[k]=1$ if blob $b_i[k]$ is assigned to track $\hat{x}_j[k]$; and $a_{ij}[k]=0$ otherwise. In k-th frame there are M[k] blobs extracted to be assigned, $b[k]=\{b_1[k],\ldots,b_{Mk}[k]\}$, and the objects tracked up to them (last assignment of blobs was at frame k-1) are: $X[k-1]=\{O_1[k-1],\ldots,O_{Nk-1}[k-1]\}$.
The size of matrix A[k], (1+N[k])xM[k], changes with time, since i=1,…,M[k] represents the blobs extracted from the k-th frame, whose number depends on the variable effects mentioned above during the detection process. Furthermore, N[k] represents the objects in the scene, whose number may also dynamically change when objects appear and disappear in the scene. Special case j=0 is considered to represent assignment of blobs to "null track" at current time, which are used to initialize new objects or are discarded.

5. Multi-blob data association with EDAs

The association problem has been defined as a search over possible blob assignments. This problem could be defined as minimizing a heuristic function to evaluate blob assignments by an efficient algorithm (Estimation of Distribution Algorithm). The heuristic function takes a Bayesian approach to model the errors in observations. The formulation of data association as a minimization problem solved by a genetic technique is not a handicap with respect to the required operation in real time. A worst-case number of operations can be fixed and bound the time consumed by the algorithm, if we restrict the maximum number of evaluations. Then, given a certain population size, the algorithm will run a number of generations limited by this bound on the number of evaluations. The most important aspect is that the EDA should converge to acceptable solutions with these conditions of limited population size and number of generations.

5.1 Encoding and efficient search with EDA algorithms

In the mathematical formulation defined in section 2, the association consists of finding the appropriate values for assignment matrix **A**, where element **A**(i, j) is 1 if blob i is assigned to object j and 0 in the opposite case. In order to be able to use the techniques of evolutionary algorithms, the matrix **A** is codified in a string of bits, being the size of matrix **A** NxM, with N the number of extracted blobs and M the number of objects in the scene. A first possibility for problem encoding was tried with a string of integer numbers representing the possible M objects to be assigned for each blob, including the "null" track 0, as shown in Figure 4:

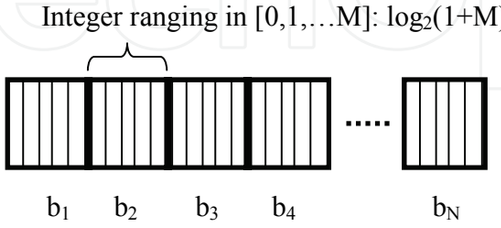


Fig. 4. Simple encoding for blob assignment

This encoding requires strings of $N\log_2(1+M)$ bits and has the problem of constraining search to solutions in which each blob can belong to one object at most. This could be a problem in situations where images from different objects get overlapped and may leave some tracks unassigned and lost.

Then, a direct encoding of A matrix was used for general solutions, where the positions in the string represent the assignations of blobs to tracks. With this codification, where individuals need $N(1+M)$ bits, a blob can be assigned to several objects:

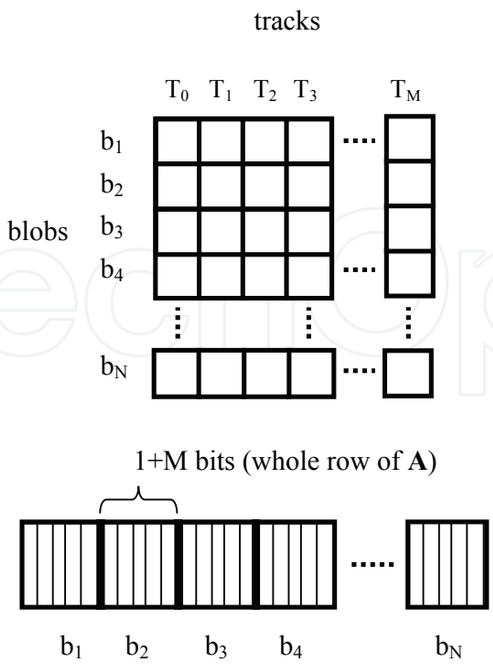


Fig. 5. Direct encoding for whole A matrix

Finally, in order to allow and effective search, the initial individuals are not randomly generated but they are fixed to solutions in which each blob is assigned to the closest object. So, the search is performed over combinations starting from this solution in order to optimize the heuristic after changing any of this initial configuration. Besides, for the case of EDA algorithms, the vector probabilities are constrained to be zero for the case of very far pairs (IF dCentroid(i,j)>GATE_THRESHOLD0=> p_{ij}=0) and those blobs which fall in spatial gates of more than one track have a non-zero change probability.

5.2 Description of the fitness function

Fitness function supplies to the evolutionary algorithms with a score to evaluate the assignment of grouping blobs to active tracks. In this case, the potential assignments explored by EDA algorithms are qualified following the heuristic described in this section. The tracking system keeps information about the centroid position, rectangle bounds and velocity for each track by means of a Kalman filter updating a track state vector, $\hat{x}^i[k]$, adapted to the dynamics of interesting objects. Therefore, we are able to evaluate the similarity between the potential assignment explored by evolutionary algorithm and the prediction of target accordingly to Kalman filter in every frame. Moreover, the fitness function should consider those assignments that leave confirmed tracks with no updating blobs.

Let $O_i[k+1]$ be the set of blobs assigned to track *j* by an individual in evolutionary algorithm for k+1 frame. This set of blobs are represented by its bounding box, specifically by its centroid pixel coordinates, width and height (x_j^e, y_j^e, w_j^e and h_j^e). They are those blobs corresponding to indexes i such that **A**ij[k+1]=1. Track *j* contains the prediction provided by Kalman filter, $\hat{x}_j[k+1|k]$, represented by its centroid pixel coordinates, width and height i (x_j^u, x_j^u, w_j^u and h_j^u).

Let d_j be the normalized distance between the evolutionary proposal and predicted track *j*:

$$d_j = \frac{\left|x_j^u - x_j^e\right|}{w_j^u} + \frac{\left|y_j^u - y_j^e\right|}{h_j^u} \tag{9}$$

Let s_j be the normalized size similarity between the evolutionary proposal and predicted track *j*:

$$s_j = \frac{\left|w_j^u - w_j^e\right|}{w_j^u} + \frac{\left|h_j^u - h_j^e\right|}{h_j^u} \tag{10}$$

Let I be the foreground image that we are processing. We define $I(x,y)$ as true, if and only if the pixel (x,y) is in the bounding box of the evolutionary proposal for track *j*. We define the Density Ratio, dr_j, of track *j* as:

$$dr_j = \frac{Card\{I(x,y)\}}{x_j^e * y_j^e} \tag{11}$$

The fitness function, F_i, of an assignment *i* that we have to minimize is:

$$F_i = \sum_{j=1}^{M} (d_j + s_j - dr_j) + ap + lt \tag{12}$$

where, M represents the number of tracks in the frame. So, we incorporate besides two penalization ratios to the fitness function, corresponding to the probabilities of false positive detection (PFA), and probabilities of true positive missing (PD):

- *ap* (assignment penalization). A penalization value is added to the fitness function every time a blob is assigned to a distant track.
- *lt* (lost track). A high value is added every time no blob is assigned to one track.

6. Results

In this section we present the comparison between EDA algorithms presented in this work and a benchmark algorithm based on a combination of Particle Filter [16], [17] and Mean Shift [18], [19] algorithms.

6.1 Video data set definition

The datasets used throughout this paper are employed with three different scenes. These datasets are from two different sources: the publicly available CVBASE dataset [2] and a DV camcorder. The datasets are quite diverse in their technical aspects and the quality of the image sequences radically differ from poor to excellent along with their pixel resolutions.

The following will describe the 3 datasets to gain an understanding of its scene characteristics:

- **Maritime scenes (BOAT).** The videos were recorded in an outdoor scenario using a DV videorecorder. The videos have a high quality with a resolution of 720x480 pixels with 15 fps. The videos feature several boats in an outdoor environment lit by the sun. The videos are very interesting due to the complex segmentation of maritime scenes. The sea has continuous movement, which contributes to the creation of a great amount of noisy blobs.
- **Squash tournament (SQUASH).** The videos are from the CVBASE dataset and were taken on a tournament of recreative players. The videos were recorded in S-VHS videorecorder, using a birds-eye view with wide angle lens. The videos were digitized to digital video format with 25 fps, resolution 384x576 and M-JPEG compression. The selected video is a zenithal record of two players playing squash. They are with close proximity to each other, similar modality of dress, slightly faster movements and constant crossings between players, which make for a challenging sequence.
- **Handball match (HANDBALL).** The videos are also from the CVBASE dataset and have the same characteristics than the squash tournament sequences. Players do not leave the court during the match, there are constant crossings among players with occlusions and disocclusions and the number of objects (players) to track is quite high. These conditions make also for a challenging sequence.

6.2 Evaluation metric definition

Before comparing the obtained results of algorithms used for visual tracking accomplishment, the first step is to determine the metric that allows making the comparisons of the algorithms behaviour. A specific quality measurement of the association has not been used and the global behaviour has been observed. In the analysis, a special emphasis in the speed of algorithms convergence is made in order to evaluate the application of the proposal development in real-time video tracking. The measures taken into account are: Tracks per Frame, Frames per Second and Time per Track.

Measurement TPF (Tracks per Frame) is used to compare the behaviour of the tracking algorithms in terms of continuity of the tracks. An optimal tracker would have to obtain the value referred as "ideal". When the obtained value is below to this "ideal" value, it means the tracker lost in the continuity of the tracks (merge effect) and, conversely, when it is over of "ideal" value, the tracker had an excess of tracks (split effect). The standard deviation of TPF allows discriminating between the behaviours with very similar averages but worse quality (greater deviation). FPS (Frames per Second) is the rate of processed images having applied the tracking algorithms; high values imply a capacity of greater processing. Column TPT (Time per Track) shows the time in milliseconds that the updating algorithm of tracks uses in the association logic, in this case the association is solved by EDAS and GA. The particles filter algorithm incorporates its own strategy of association. The rest of the time necessary to make the tracking (detection and filtrate) is common for all and therefore is not compared.

Besides, in order to grasp the algorithms' behaviour regarding convergence, bi-dimensional histograms with the number of evaluations necessary to obtain the solution and final fitness values are also presented. They have been computed depending on the size of combinatorial search space of data association hypotheses. This size is given, accordingly to encoding in section 4.4 and for each frame processed, by $2^{N(1+M)}$, being N the number of blobs and M the number of active tracks. The relative frequencies are indicated with levels of grey: black is 100%, white is 0%. It is expected that the size of search space makes more difficult the convergence, requiring more evaluations and/or converging to worse solutions.

6.3 Results and discussion

In the following tables the quality measurements of the EDAs, GA and particles filter applied to BOAT, SQUASH and HANDBALL sequences are presented. The parameters of the algorithms, size of the population, number of iterations, rate of variation, etc. have been set corresponding to the problem KP0/1 of length 20.

Recording BOAT displays a scene in the sea with three objects that remain visible in all the frames. Table 2 shows the values of the quality parameters and standard deviations obtained for this scenario.

	mean TPF (ideal=3)	sd TPF	FPS	mean TPT (millisecond)	sd TPT
CGA	2.98	0.07	4.29	2.22	0.08
UMDA	2.93	0.17	4.26	4.16	0.75
PBIL	2.98	0.07	4.04	9.14	1.93
MSPF	2.98	0.07	2.33	67.93	0.78
GA	2.93	0.17	2.21	79.37	18.23

Table 2. Measures of quality of the algorithms applied to BOAT

The results obtained for sequence BOAT show clearly the advantage of the EDAS on GA and particle filter. Observing the referring columns TPF, the quality of the tracking is the same for all the algorithms, very close to the ideal value. UMDA and GA are slightly less stable compared to the rest, but the difference is negligible. The speed that the EDAS solve the combinatorial problem of the association of blobs to tracks is quite superior (50%) to GA and particles filter.

In the following test video, SQUASH, given the normal dynamics of game, there are many situations in which the players move very close and with abrupt movements, which suppose an increase of the complexity of the association problem. The results of the quality measures are in the following table.

	mean TPF (ideal=2)	**sd TPF**	**FPS**	**mean TPT (millisecond)**	**sd TPT**
CGA	1.88	0.24	14.22	0.78	0.15
UMDA	1.87	0.25	14.40	1.04	0.18
PBIL	1.89	0.23	12.31	4.02	1.48
MSPF	1.90	0.24	5.74	53.22	2.86
GA	1.87	0.26	6.64	37.65	13.58

Table 3. Measures of quality of the algorithms applied to SQUASH

The quality of the tracking has descended due to the increase in the complexity of the scene. The best quality is obtained with the particle filter but all the values are very close. The required time to process the scene continues being very advantageous for the EDAS, superior to the double that the time obtained with the particle filter. Again, have been used the parameters of the algorithms (see Table 1) that were fit when solving the problem of the KP0/1 of length 20.

Finally we show the results on an extraordinarily complex scene. A zenithal camera records a handball match (HANDBALL). In the sequence, there are seen 14 players and 2 referees. Due to the great number of players the accumulations of several of them in small regions of the field are frequent. And the size of space search is in the best case, when only there is a blob to assign by track, of $2^{14x14} = 2^{196}$ different hypotheses. Compared with the previous scenes this is 100 greater orders of magnitude. In this case, the parameters of the algorithms corresponding to the problem of the KP0/1 of length 40 have been applied.

	mean TPF (ideal=16)	**sd TPF**	**FPS**	**mean TPT (millisecond)**	**sd TPT**
CGA	10.77	1.10	0.81	109.59	9.78
UMDA	7.35	1.01	0.31	1554.12	974.84
PBIL	11.64	1.22	0.66	17.20	13.23
MSPF	12.92	0.52	0.56	160.18	1.21
GA	12.40	1.82	0.04	43202.27	9343.08

Table 4. Measures of quality of the algorithms applied to HANDBALL

In Table 4 it can be noticed that the reduction in the quality of the tracking is very high. The value obtained in FPS implies that these algorithms cannot be used in these conditions to make real-time video tracking.

7. Conclusions

In this chapter, the association problem for real-time tracking in video was formulated as search in a hypotheses space. It is defined as a combinatorial problem, constraining the computational load to allow image processing in real time of the sequence of frames.
Evolutionary Computation techniques have been applied for solving this search problem, in particular Estimation Distribution Algorithms (EDA) that shows an efficient computational behaviour for real-time problems. The authors have done an exhaustive analysis of EDAs algorithms using several KP0/1 problems. These experimentations help the authors to know the complexity degree of the association problem and to find out the most suitable parameters for real-time video tracking problem.
From the parameters obtained in analyzing the KP0/1 problems, the author have been carried out a wide comparison among standard Genetic Algorithm, Particle Filtering based on Mean-Shift weight and several EDA algorithms: CGA, UMDA and PBIL. Three video recordings of different complexity and problematic characteristics have been used to analyze the algorithms performance. Results show the efficiency of EDA algorithms to solve the combinatorial problem in real time and the capacity to be applied in video tracking systems.

8. Acknowledgements

This work was supported in part by Projects CICYT TIN2008-06742-C02-02/TSI, CICYT TEC2008-06732-C02-02/TEC, SINPROB, CAM MADRINET S-0505 /TIC/0255 and DPS2008-07029-C02-02.

9. References

[1] M.A. Patricio, Castanedo, F., Berlanga, A., Perez, O., Garcia, J., & Molina, J. (2008). Computational Intelligence in Multimedia Processing: Recent Advances, chap. Computational Intelligence in Visual Sensor Networks: Improving Video Processing Systems, pp. 351-377. Springer Verlag.

[2] Machine Vision Group, University of Ljubljana: Cvbase ’06 workshop on computer vision based analysis in sport environments, found at url: http://vision.fe.uni-lj.si/cvbase06/ (2006)

[3] P. Larrañaga and J. A. Lozano, “Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation”. Kluwer Academic Publishers, 2001.

[4] M. Pelikan, D. E. Goldberg and E. Cantú-paz, “Linkage Problem, Distribution Estimation and Bayesian Networks”, *Evolutionary Computation* 8(3):311-340, 2000.

[5] P. Bosman, T. Alderliesten, “Evolutionary Algorithms for Medical Simulations – A Case Study in Minimally–Invasive Vascular Interventions”, in *Proc. of the 7th Int. Conf. on Genetic and Evolutionary Computation* GECCO '05, 2005.

[6] H. Mühlenbein, “The equation for response to selection and its use for prediction”, *Evolutionary Computation*, 5:303–346, 1998.

[7] S. Baluja, “Population based incremental learning: A method for integrating genetic search based function optimization and competitive learning”, Technical Report No. CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1994.

[8] G. Harik, F. Lobo and D. Goldberg, "The Compact Genetic Algorithm", *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, 1999.

[9] B. Cestnik, "Estimating probabilities: A crucial task in machine learning", in L. C. Aiello, ed. *Proc. of the 9th European Conference on Artificial Intelligence*, pp. 147-149. Pitman Publishing, 1990.

[10] D. Pisinger, "Where are the hard knapsack problems?", *Computers & Operations Research*, vol. 32, issue 9, pp. 2271-2284, 2005.

[11] R. Merkle and M. Hellman, "Hiding and signatures in trapdoor knapsacks", *IEEE Trans. Information Theory*, 24:525-530, 1978.

[12] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. "Multicamera People Tracking with a Probabilistic Occupancy Map" IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 30, NO. 2, FEBRUARY 2008, pp 267-282

[13] C. Rasmussen and G. D. Hager, "Probabilistic data association methods for tracking complex visual objects," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 6, pp. 560–576, June 2001.

[14] I. J. Cox and S. L. Hingorani. "An efficient implementation of reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. IEEE Transaction on Pattern Analysis and Machine Intelligence, 18(2) Feb 1996. pp 138–150.

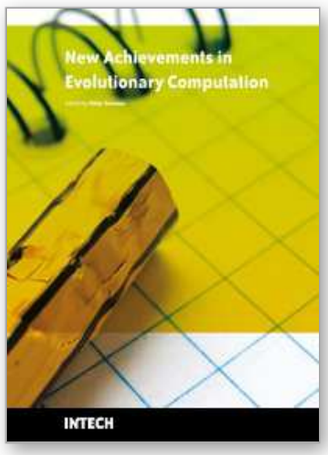
[15] A. Loza A, Miguel A. Patricio M A, J. Garcia J, J.M. Molina. "Advanced algorithms for real-time video tracking with multiple targets". 10th International Conference on Control, Automation, Robotics and Vision, ICARCV 2008. Hanoi, Vietnam. Dec 17-20, 2008.

[16] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking", *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.

[17] S. A. Branko Ristic and N. Gordon, "Beyond the Kalman Filter: Particle Filters for Tracking Applications", Artech House Publishers, 2004.

[18] D. Comaniciu, V. Ramesh and P. Meer, "Real-time tracking of non-rigid objects using mean shift" in US Patent and Trademark Office, 2000.

[19] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 603-619, 2002.



New Achievements in Evolutionary Computation

Edited by Peter Korosec

ISBN 978-953-307-053-7
Hard cover, 318 pages
Publisher InTech
Published online 01, February, 2010
Published in print edition February, 2010

Evolutionary computation has been widely used in computer science for decades. Even though it started as far back as the 1960s with simulated evolution, the subject is still evolving. During this time, new metaheuristic optimization approaches, like evolutionary algorithms, genetic algorithms, swarm intelligence, etc., were being developed and new fields of usage in artificial intelligence, machine learning, combinatorial and numerical optimization, etc., were being explored. However, even with so much work done, novel research into new techniques and new areas of usage is far from over. This book presents some new theoretical as well as practical aspects of evolutionary computation. This book will be of great value to undergraduates, graduate students, researchers in computer science, and anyone else with an interest in learning about the latest developments in evolutionary computation.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Antonio Berlanga, Miguel A. Patricio, Jesús García, and José M. Molina (2010). Solving Combinatorial Problems with Time Constrains Using Estimation of Distribution Algorithms and Their Application in Video-Tracking Systems, New Achievements in Evolutionary Computation, Peter Korosec (Ed.), ISBN: 978-953-307-053-7, InTech, Available from: http://www.intechopen.com/books/new-achievements-in-evolutionary-computation/solving-combinatorial-problems-with-time-constrains-using-estimation-of-distribution-algorithms-and-

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License, which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen