

Estimation of Distribution Algorithm based on Hidden Markov Models for Combinatorial Optimization

Marc-André Gardner, Christian Gagné, and Marc Parizeau

Laboratoire de vision et systèmes numériques

Département de génie électrique et de génie informatique

Université Laval, Québec (Québec), Canada G1V 0A6

marc-andre.gardner.1@ulaval.ca, {christian.gagne, marc.parizeau}@gel.ulaval.ca

ABSTRACT

Estimation of Distribution Algorithms (EDAs) have been successfully applied to a wide variety of problems. The algorithmic model of EDA is generic and can virtually be used with any distribution model, ranging from the mere Bernoulli distribution to the sophisticated Bayesian network. The Hidden Markov Model (HMM) is a well-known graphical model useful for modelling populations of variable-length sequences of discrete values. Surprisingly, HMMs have not yet been used as distribution estimators for an EDA, even though it is a very powerful tool especially designed for modelling sequences. We thus propose a new method, called HMM-EDA, implementing this idea. Preliminary comparative results on two classical combinatorial optimization problems show that HMM-EDA is indeed a promising approach for problems that have sequential representations.

Categories and Subject Descriptors

I.2.8 [Artificial Intelligence]: Problem Solving, Control Methods, and Search—*Heuristic methods*; G.3 [Probability and Statistics]: *Markov processes*

Keywords

Estimation of distribution algorithms; Hidden Markov models; Combinatorial optimization

Several graphical models have been used in EDAs. The most widely known is probably the Bayesian optimization algorithm (BOA) [3], which relies on a Bayesian network to learn the model and sample solutions, but there are more, like DEUM (Distribution Estimation Using Markov network) [5] and dtEDA (dependency-tree EDA) [4]. HMM-EDA is a novel approach which uses a Hidden Markov Model (HMM) [2] as the underlying distribution. While its versatility may allow it to model various kinds of distributions, the focus of this work is on combinatorial optimization problems.

The proposed approach integrates an HMM in the generic EDA model, using it to directly estimate the distribution of the samples making up the population. Using an HMM for that purpose is quite versatile, as it only assumes that the samples are sequences of discrete values. Such a model should be useful in a wide variety of optimization problems, including problems where solutions can be modelled

as bit strings. HMM should be able to capture complex interactions between the elements of sequences, not only the first order relations captured in observable Markov models. Our intuition is that hidden states of the HMM can capture intrinsic sequential patterns that characterize populations, and link these patterns together in various ways to produce better individuals, through transitions between the hidden states.

The HMM-EDA algorithm interprets individuals as sequences. Each possible value of the problem's "alphabet" is mapped to a possible observation of the HMM. For instance, with the travelling salesman problem, there would be as many possible observations as the number of cities in the tour. Given that HMMs can easily handle up to several dozens of observable emissions, the implementation is quite straightforward for many problems. The HMM is trained with the best individuals, and then produces a new generation, effectively leading to an offspring population with, hopefully, a better average fitness than the parent population. In the process, the probabilistic nature of the HMM will allow for exploring some regions of the search space. A detailed description of the method based on a $(\mu + \lambda)$ selection loop is presented in Algorithm 1.

Some problems require that the generation of the same observation twice in a given individual be avoided. In order to do so, restrictions can be applied on the emitted observations, to ensure that only valid individuals are generated. This restriction is implemented by adding a binary mask over the emission matrix. At emission time, the probability of a used emission is set to 0, and the other probabilities are normalized to 1.

The proposed method also makes use of some mutation when generating individuals with the HMM. It consists in modifying the usual mechanism of observation emission, such that, according to a given mutation probability (about 2%), an observation is selected randomly (uniformly) instead of using the HMM emission probabilities of the current state.

Experiments were made using two classical problems: the *Travelling Salesman Problem* (TSP), and the *0-1 Knapsack*. Comparison is made with a standard GA (using a permutation representation), and, in the case of Knapsack, with the PBIL EDA [1]. For each problem, 100 independent runs were conducted. The stopping criteria used was the number of evaluations performed. The population was of $(\mu + \lambda) = (250 + 750)$ for HMM-EDA and 500 for GA and PBIL. A tournament size of 5 was used for selection, and the number of hidden states was set to 3 for the Knapsack and 25 for the TSP. All experiments were conducted using

Algorithm 1 Optimization loop of HMM-EDA.

```

1:  $P^0 \leftarrow \text{initialize\_population}(\mu)$ 
2: Estimate fitness of all individuals in  $P^0$ 
3: for  $i = 1 \rightarrow i_{\max}$  do
4:    $P_{\text{train}} \leftarrow \text{tournament}(P^{i-1}, n)$ 
5:    $\theta_0 \leftarrow$  initial random HMM parameters
6:    $\theta \leftarrow \text{BaumWelch}(P_{\text{train}}, \theta_0)$ 
7:    $P_{\text{offspring}} = \emptyset$ 
8:   for  $k = 1 \rightarrow \lambda$  do
9:     Reset HMM state
10:     $t \leftarrow \text{generate\_offspring}(\theta)$ 
11:     $P_{\text{offspring}} \leftarrow P_{\text{offspring}} \cup \{t\}$ 
12:   end for
13: Estimate fitness of all individuals in  $P_{\text{offspring}}$ 
14:  $P_{\text{offspring}} \leftarrow P_{\text{offspring}} \cup P^{i-1}$ 
15:  $P^i = \emptyset$ 
16: repeat
17:    $t \leftarrow \text{select\_best}(P_{\text{offspring}})$ 
18:   if not  $t \in P^i$  then
19:      $P^i = P^i \cup \{t\}$ 
20:   end if
21:    $P_{\text{offspring}} = P_{\text{offspring}} \setminus \{t\}$ 
22: until  $|P^i| = \mu$  or  $|P_{\text{offspring}}| = \emptyset$ 
23: end for

```

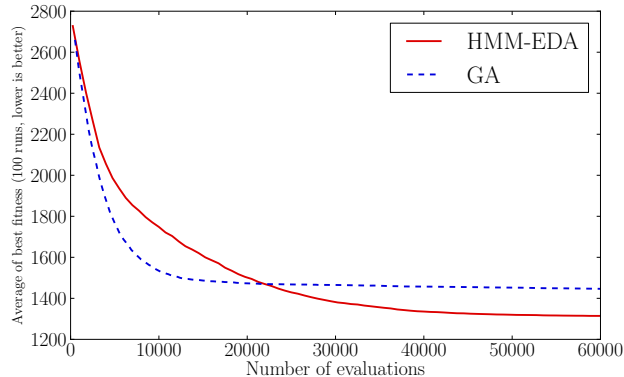


Figure 1: Evolution of the fitness against the number of evaluations performed for TSP.

the DEAP (Distributed Evolutionary Algorithms in Python) framework¹.

Figures 1 and 2 plot the best fitness against the number of evaluations performed, while Figure 3 presents a boxplot of the final best-of-run fitness for each problem.

Results show that HMM-EDA always offers a respectable level of performance when compared to the permutation representation GA approach. Indeed, there is no situation where the HMM falls far behind the GA, whether we look at the final performance or the convergence speed.

On the specific case of the Knapsack problem, results unveil that HMM-EDA reaches its optimum value about two times faster than the GA, which is an interesting achievement. The experiments with TSP show another scenario, where GA converges faster than HMM-EDA, but with a significantly worse final fitness value. Moreover, the standard deviation on the HMM-EDA results is almost one third of the GA, meaning that the approach is quite stable.

All in all, experiments confirmed some promising features

¹<http://deap.gel.ulaval.ca>

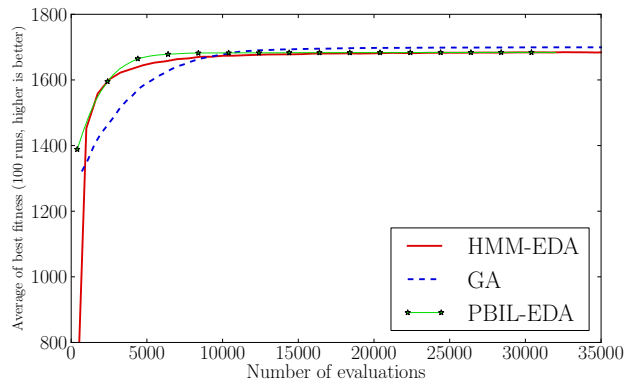


Figure 2: Evolution of the fitness against the number of evaluations performed for 0-1 Knapsack.

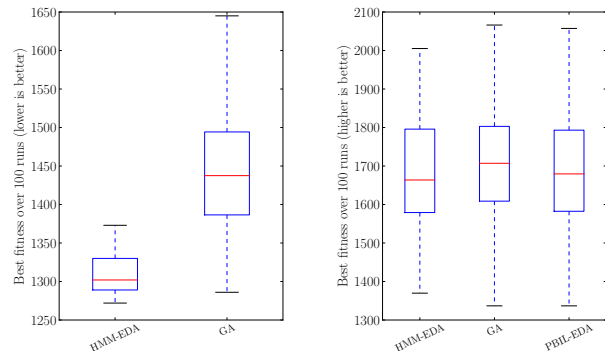


Figure 3: Box plots of the final best fitness (left : TSP, right : 0-1 Knapsack)

of HMM-EDA, both from fitness and performance viewpoints, and suggest that other studies should be conducted to unveil its full potential, in particular with other sequential or combinatorial problems.

Acknowledgements

This work was made possible through funding from NSERC (Canada) and access to computational resources of Calcul Québec / Compute Canada.

1. REFERENCES

- [1] S. Baluja. Population-based incremental learning: a method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, PA, 1994.
- [2] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.
- [3] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. BOA: The Bayesian Optimization Algorithm. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO)*, 1999.
- [4] M. Pelikan, S. Tsutsui, and R. Kalapala. Dependency trees, permutations, and quadratic assignment problem. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO)*, 2007.
- [5] S. Shakya, A. Brownlee, J. McCall, F. Fournier, and G. Owusu. A fully multivariate DEUM algorithm. In *IEEE Congress on Evolutionary Computation (IEEE-CEC)*, 2009.