Selection and Reinforcement Learning for Combinatorial Optimization (updated version)

A. Berny

IRIN, Université de Nantes, France
berny@irin.univ-nantes.fr

Abstract. Improving on a previous paper, we explicitly relate reinforcement and selection learning (PBIL) algorithms for combinatorial optimization, which is understood as the task of finding a fixed-length binary string maximizing an arbitrary function. We show the equivalence of searching for an optimal string and searching for a probability distribution over strings maximizing the function expectation. In this paper however, we will only consider the family of Bernoulli distributions. Next, we introduce two gradient dynamical systems acting on probability vectors. The first one maximizes the expectation of the function and leads to reinforcement learning algorithms whereas the second one maximizes the logarithm of the expectation of the function and leads to selection learning algorithms. We finally give a stability analysis of solutions.

1 Introduction

Statistical approaches for optimization and learning have been used for many years. Only in the restricted case of Bernoulli distributions, one can cite Stochastic Learning Automata from Narendra and Thathachar [9], Associative Reward Penalty from Barto and Anandan [2], REINFORCE from Williams [13], BSC from Syswerda [12], Population-Based Incremental Learning from Baluja and Caruana [1], Relative Payoff Procedure from Dayan and Hinton [5] and Univariate Marginal Distribution from Mülhenbein [8]. All these algorithms update a probability vector to increase the expected reward, that is the expectation of the function to be maximized relatively to the Bernoulli distribution given by the probability vector. Algorithms using more complex probability distributions, such as those exhibiting chain or tree-like dependency graphs, have been proposed to overcome the limitations of those based on Bernoulli distributions (see [7] for a review). However, those limitations are not clearly identified. What is known is that Bernoulli algorithms perform better than hill climbers at a relatively low computational cost.

In this paper, we analytically study Bernoulli algorithms seen as gradient dynamical systems, putting the stress on their local behavior. We extend the work presented in [3] and apply some of the ideas of [?] in the context of binary string search.

In Section 2, we introduce a statistical framework for optimization over fixed-length binary strings. We show that maximizing the expectation of the function to be maximized is equivalent to maximizing the function itself. In Section 3, we introduce two equivalent gradient dynamical systems acting on probability vectors, leading respectively to reinforcement and selection learning (PBIL). Section 4 gives a Liapunov stability analysis of exact strings which relies on combinatorial techniques. Section 5 is an introduction to other stability issues.

2 Statistical Framework

2.1 Introduction and Notations

The problem is to maximize an arbitrary function over the set of fixed-length binary strings. We recast this problem as the problem of finding a distribution over binary strings maximizing some statistical criterion. The criterion is the expectation of the function relatively to the distribution. We will show that both problems are equivalent.

Let $S = \{0,1\}^n$ be the set of binary strings of length n. We will also refer to S as the n-dimensional hypercube. Let $f : S \to \mathbf{R}$ be the function to maximize. We require that f be positive, that is $f \geq a > 0$. Let $M(S)$ be the set of distributions over S and $J : M(S) \to \mathbf{R}$ the statistical criterion defined for each $\mu \in M(S)$ as $J(\mu) = \boldsymbol{E}_\mu(f) = \sum_{x \in S} \mu(x) f(x)$. From the positivity of f, we have $J \geq a > 0$.

Let $U = \{x \in S : f(x) = f^*\}$, where $f^* = \sup_S f$, and $V = \{\mu \in M(S) : J(\mu) = \sup_{M(S)} J\}$ be the optimal sets for f and J respectively. Let δ_a be the Dirac distribution charging point $a \in S$, that is $\delta_a(x) = 1$ if $x = a$ and $\delta_a(x) = 0$ otherwise.

2.2 Equivalence of Problems

We first observe that $J(\mu) \leq f^*$ and that $J(\delta_x) = f^*$ for all $x \in U$. Consequently, $\{\delta_x : x \in U\} \subset V$. We have the following theorem :

Theorem 1. *$\mu \in V$ if and only if $\mu(x) = 0$ for all $x \in S \backslash U$ or, put another way, μ is a mixture of Dirac distributions charging points in U.*

Proof. Let $\mu \in V$. Then

$$\begin{aligned} J(\mu) &= f^* \\ &= \sum_{x \in U} \mu(x) f(x) + \sum_{x \in S \backslash U} \mu(x) f(x) \\ &= f^* \sum_{x \in U} \mu(x) + \sum_{x \in S \backslash U} \mu(x) f(x). \end{aligned}$$

Hence,

$$f^* \left(1 - \sum_{x \in U} \mu(x)\right) = \sum_{x \in S \setminus U} \mu(x) f(x)$$
$$f^* \sum_{x \in S \setminus U} \mu(x) = \sum_{x \in S \setminus U} \mu(x) f(x).$$

Suppose there exists an $x \in S \setminus U$ such that $\mu(x) > 0$. We divide both sides of the above equation by $\sum_{x \in S \setminus U} \mu(x)$ and obtain

$$f^* = \sum_{x \in S \setminus U} \left(\frac{\mu(x)}{\sum_{x \in S \setminus U} \mu(x)} \right) f(x).$$

f is upper bounded over $S \setminus U$ by $\sup_{S \setminus U} f < f^*$. We finally reach the contradiction $f^* \leq \sup_{S \setminus U} f < f^*$ and $\mu(x) = 0$ for all $x \in S \setminus U$. Conversely, if $\mu(x) = 0$ for all $x \in S \setminus U$ then $J(\mu) = \sum_{x \in U} \mu(x) f^* = f^*$.

2.3 Bernoulli Distributions

We restrict the search for an optimal distribution to a simple family of distributions, the Bernoulli distributions. More precisely, we make the assumption of independent random variables for the components of binary strings.

In order to express Bernoulli distributions, we introduce the function $\phi : \{0,1\} \times [0,1] \to [0,1]$ defined by

$$\phi(x_i, v_i) = \begin{cases} v_i & \text{if } x_i = 1 \\ 1 - v_i & \text{if } x_i = 0 \end{cases}$$

which can be interpreted as the probability of the ith component x_i of binary string $x = (x_1, \ldots, x_n) \in S$. Let Δ be the box $[0,1]^n$ whose vertices are the points of S. $v = (v_1, \ldots, v_n) \in \Delta$ is called a probability vector. We define the Bernoulli distribution μ_v associated to the probability vector v as

$$\mu_v(x) = \prod_{i=1}^{n} \phi(x_i,\, v_i).$$

Observe that if $v \in S$ then $\mu_v = \delta_v$.

3 Dynamical System Approach

3.1 Reinforcement Learning

We search for dynamical systems acting on probability vectors of Bernoulli distributions in order to maximize the statistical criterion J. We first turn to gradient techniques and introduce the dynamical system

$$v' = \operatorname{grad} J, \tag{1}$$

where v' is the time derivative of the solution $v(t)$. However, (1) does not constrain v to stay in Δ, which is the condition to interpret v as a probability vector. Thus, we stabilize (1) with the poles 0 and 1, by setting, if $v' = \varphi(v)$ and $\varphi = (\varphi_1, \ldots, \varphi_n)$,

$$\varphi_i = v_i(1 - v_i) \frac{\partial J}{\partial v_i}. \tag{2}$$

Let Ω be the interior of Δ, that is $\Omega = (0,1)^n$. By continuity arguments, if $v(0)$ is in Ω, so is $v(t)$ for all $t \in \mathbf{R}$. In most cases, one chooses $v(0) = (1/2, \ldots, 1/2)$, which reflects the lack of a priori information on the problem.

We relate the polynomial $v_i(1 - v_i)$ to the derivative of the logistic function which is a sigmoid function. Let $g : \mathbf{R} \to (0,1)$ be the logistic function defined as

$$g(x) = \frac{1}{1 + e^{-x}}.$$

We have the relation $g' = g(1 - g)$. This suggests the introduction of a vector $\theta \in \mathbf{R}^n$, with $\theta = (\theta_1, \ldots, \theta_n)$, whose components are defined by $\theta_i = g^{-1}(v_i)$. The dynamical system acting on θ is simply

$$\theta_i' = \left. \frac{\partial J}{\partial v_i} \right|_{g(\theta)}.$$

In order to calculate $\partial J / \partial v_i$, we first exchange the derivative and the sum over S, which gives

$$\frac{\partial J}{\partial v_i} = \sum_{x \in S} \frac{\partial \mu_v}{\partial v_i} f(x)\,. \tag{3}$$

We write

$$\frac{\partial \mu_v}{\partial v_i} = \frac{\partial \log \mu_v}{\partial v_i} \times \mu_v(x)$$

which enables us to interpret the sum over S in (3) as an expectation. We finally write the gradient of J like

$$\boldsymbol{E}\big(e(x_i, v_i) f(x)\big)\,, \tag{4}$$

where

$$e(x_i, v_i) = \frac{\partial \log \mu_v(x)}{\partial v_i} = \begin{cases} -1/1 - v_i & \text{if } x_i = 0 \\ 1/v_i & \text{if } x_i = 1 \end{cases}$$

is called the eligiblity in the context of reinforcement learning.

Using a stochastic approximation with a comparison scheme, we obtain the following classical REINFORCE algorithm [13]

$$\theta_i(t+1) = \theta_i(t) + \alpha\, e(x_i, v_i) \Big(f(x(t)) - \boldsymbol{E}(f) \Big),$$

where $\alpha > 0$ is the learning rate and $x(t)$ is the sample generated at time t from the distribution μ_v. The algorithm we have presented in [3] is slightly different.

3.2 Selection Learning

Proportional selection Instead of a gradient dynamical system maximizing J, we consider a gradient dynamical system maximizing $\log J$ whose vector field is given by

$$\varphi_i = v_i(1-v_i)\frac{\partial \log J}{\partial v_i}\,. \tag{5}$$

Using expression (4) for the gradient of J and the simple relation

$$\frac{\partial \log J}{\partial v_i} = \frac{1}{J} \times \frac{\partial J}{\partial v_i}$$

we find

$$v_i' = \boldsymbol{E}\Big((x_i - v_i) \times \frac{f(x)}{\boldsymbol{E}(f)}\Big)\,. \tag{6}$$

Expressions (2) and (5) define the same field of directions and the resulting dynamical systems are thus equivalent.

We interpret the right-hand side of (6) as the effect of a fitness proportional selection operator, which is allowed by the fact that f is positive. Let us introduce the distribution ν which is the image of μ_v under the transformation

$$\mu_v(x) \longrightarrow \nu(x) = \frac{f(x)\mu_v(x)}{\boldsymbol{E}(f)}.$$

We can think of ν as an infinite population resulting from the application a fitness proportional selection operator to the population μ_v. The operator $\boldsymbol{E}(\cdot \times f/\boldsymbol{E}(f))$ is the expectation operator $\boldsymbol{E}_\nu(\cdot)$ relatively to the distribution ν. (6) can be rewritten using the operator $\boldsymbol{E}_\nu(\cdot)$ as

$$v_i' = \boldsymbol{E}_\nu(x_i) - v_i. \tag{7}$$

The intuitive interpretation of the above equation is that v is moved towards vertices with high fitness.

Boltzmann selection We come closer to PBIL in changing the criterion and optimizing $J = \log \boldsymbol{E}(e^{\beta f})$, with $\beta > 0$, instead of $J = \log(f)$. The dynamical system can still be written like (7) but the distribution resulting from selection is obtained through the transformation

$$\mu_v(x) \longrightarrow \nu(x) = \frac{e^{\beta f(x)}\mu_v(x)}{\boldsymbol{E}(e^{\beta f})}.$$

We write the dynamical system as $v' = \varphi_\beta(v)$ and we can prove, using a Lagrange technique, that if $v \in \Omega$ then

$$\lim_{\beta\to\infty} \varphi_\beta(v) = \left(\sum_{x\in U} \frac{\mu_v(x)}{\sum_{x\in U}\mu_v(x)} x\right) - v.$$

If $U = \{x^*\}$, then the only singular point of the ideal system is x^* itself, which motivates the search for a schedule $\beta : \mathbf{N} \to \mathbf{R}_+$, with β increasing, $\beta(0) = 0$ and $\beta(\infty) = \infty$, in order to insure convergence to the global optimum x^*. Such a schedule (or annealing) has been proposed for genetic algorithms by Cerf [4].

Stochastic approximation At each iteration, N individuals (x^i), $i = 1, \ldots, N$, are sampled from μ_v which results in the empirical distribution

$$\tilde\mu = \frac{1}{N}\sum_{i=1}^{N} \delta_{x^i}\,,$$

which is then re-weighted through Boltzmann selection, yielding the new empirical distribution

$$\tilde\nu = \sum_{i=1}^{N} \frac{e^{\beta f(x^i)}}{\sum_{j=1}^{N} e^{\beta f(x^j)}} \delta_{x^i}\,.$$

If we consider the limit case $\beta \to \infty$ then $\tilde\nu$ is uniform over the best individuals in the finite population (x^i), $i = 1, \ldots, N$. If we select only one such individual x^* then v is updated with the discrete time recurrent equation

$$v(t+1) = v(t) + \alpha(x^* - v)\,,$$

where $0 < \alpha < 1$. This is very close to the basic PBIL algorithm (in [1] Baluja and Caruana use the best two vectors instead of just the best one).

4 Stability Analysis of Vertices

4.1 Introduction and Notations

In this section, we analyze the stability of points of S or vertices of the box Δ which are, by construction, singular points of (2). To do so, we study the sign of a Liapunov function [6] in the neighborhood of a vertex. We implicitly use an induced topology on Ω.

Let y be a point of S. The Liapunov function $h : \Omega \to \mathbf{R}$ is simply $h(v) = J(\mu_v) - f(y)$. Since y is in S, we have $\mu_y = \delta_y$. Then $J(\mu_y) = f(y)$ and $h(y) = 0$. The time derivative of h is

$$h' = \sum_{i=1}^{n} \frac{\partial J}{\partial v_i} v_i' = \sum_{i=1}^{n} v_i(1-v_i)\left(\frac{\partial J}{\partial v_i}\right)^2.$$

h' vanishes on y since y is a vertex. If the gradient of J does not vanish on y [1], we can find a neighborhood $W_1 \subset \Omega$ of y on which $h' > 0$. We will be able to conclude in two cases:

[1] Otherwise, in case of plateau or when $(f \circ \sigma_i)(y) = f(y)$ for all $i \in L$, we only get simple stability.

1. If we can find another neighborhood $W_2 \subset \Omega$ of y on which $h < 0$ then y is asymptotically stable;
2. If we can find another neighborhood $W_2 \subset \Omega$ of y on which h takes both signs then y is unstable.

In order to study the sign of h, we develop J in the neighborhood of y until we reach an order at which we can conclude. We first make a change of variables to handle only positive coordinates in the neighborhood of 0. Let $v = (v_1, \dots, v_n)$ be in the neighborhood of y in Ω. We define the positive vector $w = (w_1, \dots, w_n)$ such that if $y_i = 0$ then $v_i = w_i$, and if $y_i = 1$ then $v_i = 1 - w_i$. For each $x \in S$, we define the set of integers $I(x)$ such that the corresponding components of x and y are equal, that is $I(x) = \{i \in L : x_i = y_i\}$ where $L = \{1, \dots, n\}$. We also define its complementary $J(x)$. For each integer i in $I(x)$ we have $\phi(x_i, v_i) = 1 - w_i$, and for each integer i in $J(x)$ we have $\phi(x_i, v_i) = w_i$. With these notations, we have

$$\mu_v(x) = \prod_{i \in I(x)} (1 - w_i) \times \prod_{i \in J(x)} w_i \, .$$

We introduce the family of bit reversal transformations, $\sigma_i : S \to S$, $i \in L$, such that if $z = \sigma_i(y)$ then $z_i = 1 - y_i$ and $z_j = y_j$ for $j \neq i$. We use the fact that those transformations commute to write any $x \in S$ as the product $(\prod_{i \in J(x)} \sigma_i)(y)$. Summations over S can then be replaced by summations over subsets of L. We finally introduce two additional notations to facilitate reading, $w_K = \prod_{i \in K} w_i$ and $f_K = \left(f \circ \prod_{i \in K} \sigma_i\right)(y)$, where $K \subseteq L$.

4.2 Criterion Expansion

With the above notations, the criterion can be written

$$J(\mu_v) = \sum_{J \subseteq L} f_J w_J \prod_{i \in L \setminus J} (1 - w_i) \, .$$

We expand the inner product using the symmetrical polynomials

$$\chi_i(I) = \sum_{\substack{J \subseteq I \\ |J| = i}} w_J \, ,$$

and obtain

$$J(\mu_v) = \sum_{J \subseteq L} f_J w_J \sum_{i=0}^{n-|J|} (-1)^i \chi_i(L \setminus J) \, ,$$

where $|J|$ is the cardinal of J which we use to split the first sum. Then,

$$\begin{aligned}
J(\mu_v) &= \sum_{k=0}^{n} \sum_{i=0}^{n-k} (-1)^i \sum_{\substack{J \subseteq L \\ |J| = k}} f_J w_J \chi_i(L \setminus J) \\
&= \sum_{k=0}^{n} \sum_{l=k}^{n} (-1)^{l-k} \sum_{\substack{J \subseteq L \\ |J| = k}} f_J w_J \chi_{l-k}(L \setminus J) && \text{(where } l = i + k\text{)} \\
&= \sum_{l=0}^{n} (-1)^l \sum_{\substack{J \subseteq L \\ |J| \leq l}} (-1)^{|J|} f_J w_J \chi_{l-|J|}(L \setminus J) \, . && \text{(merge)}
\end{aligned}$$

We isolate the product of components and see that

$$w_J \chi_{l-|J|}(L \setminus J) = \sum_{\substack{K \subseteq L \\ J \subseteq K \\ |K| = l}} w_K \, .$$

We make the substitution into the criterion and get

$$\begin{aligned}
J(\mu_v) &= \sum_{l=0}^{n} (-1)^l \sum_{\substack{J \subseteq L \\ |J| \leq l}} \sum_{\substack{K \subseteq L \\ J \subseteq K \\ |K| = l}} (-1)^{|J|} f_J w_K \\
&= \sum_{l=0}^{n} (-1)^l \sum_{\substack{K \subseteq L \\ |K| = l}} w_K \sum_{J \subseteq K} (-1)^{|J|} f_J && \text{(exchange)} \\
&= \sum_{K \subseteq L} \alpha_K w_K \, , && \text{(merge)}
\end{aligned}$$

where $\alpha_K = (-1)^{|K|} \sum_{J \subseteq K} (-1)^{|J|} f_J$.

4.3 Sign of the Liapunov Function

First order We consider first order terms in the expansion of h which we write

$$\sum_{\substack{K \subseteq L \\ |K| = 1}} w_K (f_K - f(y)).$$

If there exists a subset K such that $|K| = 1$ and $f_K > f(y)$ then y is unstable since h and h' can take values of the same sign the neighborhood of 0.

We now consider terms of order $l \geq 2$ in the expansion of h,

$$\sum_{\substack{K \subseteq L \\ |K| = l}} \alpha_K w_K \, .$$

We say that a subset K of L has the property A_i if for each subset $J \subseteq K$ such that $|J| \leq i$, $f_J = f(y)$. We split the first sum between subsets having A_1 and those having $\neg A_1$. For subsets K having $\neg A_1$, we can find a subset $J \subseteq K$ such that $|J| = 1$ and $f_J \neq f(y)$, which we reduce to $f_J < f(y)$ since we have already seen that the reverse relation led to instability. Then J must appear in the first order terms and since w_K is an $\mathcal{O}(w_J)$ we will neglect it and simply write $\mathcal{O}(w_J, |J| = 1, f_J < f(y))$. The expansion can now be written

$$\sum_{\substack{K \subseteq L \\ |K|=1 \\ \neg A_1}} w_K(f_K - f(y)) + \mathcal{O}(w_K, |K| = 1, f_K < f(y)) + \sum_{\substack{K \subseteq L \\ |K| \geq 2 \\ A_1}} \alpha_K w_K \,.$$

Higher orders Assume that the expansion of h can be written

$$\sum_{l=1}^{p} \left(\sum_{\substack{K \subseteq L \\ |K|=l \\ A_{l-1} \wedge \neg A_l}} w_K(f_K - f(y)) + \mathcal{O}(w_K, |K| = l, f_K < f(y)) \right) + \sum_{\substack{K \subseteq L \\ |K| \geq p+1 \\ A_p}} \alpha_K w_K$$

We isolate subsets for which $|K| = p + 1$ in the remainder and simplify the corresponding term. Since those subsets have the property A_p, it is easy to show that $\alpha_K = f_K - f(y)$, which gives the first term of the expansion for the order $p+1$. Just as we have done for the first order, we split the remainder of order at least $p+2$ between subsets having A_{p+1} and those having $\neg A_{p+1}$. The latter can be neglected since they contain a subset which must appear in the first term. We have thus completed the expansion of h for the order $p + 1$. We have to decide when to stop the expansion:

1. if there exists a subset K with $|K| = p$ and $A_{p-1} \wedge \neg A_p$ such that $f_K > f(y)$ then the vertex is unstable;
2. if there exists no subset K with $|K| = p + 1$ and A_p then the vertex is asymptotically stable.

5 Other Stability Issues

Singular points of the criterion J are such that $\partial J/\partial v_i = 0$ for all $i \in L$. Let us consider the derivative $D\varphi$ of the vector field φ, which is the matrix of general element $\partial\varphi_i/\partial v_j$. Since $J(\mu_v)$ is polynomial of degree 1 in each v_i, diagonal entries in $D\varphi$ are equal to zero at singular points of J. The trace of $D\varphi$ is thus equal to zero. The trace of $D\varphi$ is also equal to the sum of its eigenvalues, which implies that if $D\varphi$ does not vanish, it has both positive and negative eigenvalues, making any singular point of J an instable point. We have not dealt with the issue of the existence of such singular points, in particular inside the hypercube.

We have neither covered the case of compound singular points such that $\partial J/\partial v_i = 0$ for some i and $v_i = 0$ or $v_i = 1$ for others nor the case of vertices which are at the same time singular points of the criterion J.

6 Conclusion

We have presented a statistical framework for combinatorial optimisation over fixed-length binary strings and shown that both selection and reinforcement learning algorithms can be derived from gradient dynamical systems acting on Bernoulli probability vectors. It should be clear that such algorithms perform as well as hill climbing, since they can only converge to locally optimal solutions. We can state that from the stability analysis of vertices, which deals with the local behavior of the system. It remains to find a criterion, geometric or algebraic, to decide whether the system will converge to the global optimum of the function. This should ultimately be related to the regularity of the function and the notion of fitness landscape [11, 10].

References

1. S. Baluja and R. Caruana. Removing the genetics from the standard genetic algorithm. In *Proceedings of the 12th Annual Conference on Machine Learning*, pages 38–46, 1995.
2. A.G. Barto and P. Anandan. Pattern recognizing stochastic learning automata. *IEEE Trans. on Systems, Man, and Cybernetics*, 15:60–75, 1985.
3. A. Berny. Statistical machine learning and combinatorial optimization. In L. Kallel, B. Naudts, and A. Rogers, editors, *Theoretical Aspects of Evolutionary Computing*, Lecture Notes in Natural Computing. Springer-Verlag, 2000.
4. R. Cerf. *Une théorie asymptotique des algorithmes génétiques*. Thèse de doctorat, université de Montpellier II, mars 1994.
5. P. Dayan and G. E. Hinton. Using EM for reinforcement learning. *Neural Computation*, 9(2):271–278, 1997.
6. M. W. Hirsch and S. Smale. *Differential equations, dynamical systems, and linear algebra*. Academic Press, 1974.
7. P. Larrañaga, R. Etxeberria, J. A. Lozano, and J. M. Peña. Optimization by learning and simulation of Bayesian and Gaussian networks. Technical Report EHU-KZAA-IK-4/99, Department of computer science and artificial intelligence, university of the Basque country, December 1999.
8. H. Mühlenbein. The equation for response to selection and its use for prediction. *Evolutionary Computation*, 5(3):303–346, 1997.
9. K.S. Narendra and M.A.L. Thathachar. *Learning automata: an introduction*. Prentice Hall, NJ, 1989.
10. B. Naudts and L. Kallel. Comparison of summary statistics of fitness landscapes. *IEEE Trans. on Evolutionary Computation*, 2000. To appear.
11. P. F. Stadler. *Complex systems and binary networks*, chapter Towards a theory of landscapes, pages 77–163. Springer-Verlag, Berlin, 1995.
12. G. Syswerda. Simulated crossover in genetic algorithms. In L.D. Whitley, editor, *Second Workshop on Foundations of Genetic Algorithms*, pages 239–255. Morgan Kaufmann, 1993.
13. R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.