# THE EXPECTED NUMBER OF STEPS IN SUCCESSIVE APPROXIMATION TYPE ALGORITHMS

By

I. Bencsik

Department of Automation, Technical University, Budapest

(Received September 13, 1973)

Presented by Prof. Dr. F. Csáki

## Introduction

The algorithms, named in the title, converge asymptotically in the stochastic case — if the convergence criteria are satisfied — with unity probability in an infinite number of steps. In the real-time case and a finite time interval the expected value of the quality criterion characterizing the convergence speed, i.e. the time behaviour of the convergence probability is questionable. In the following a possible evaluation of this time function is presented.

## Convergence Probability Versus Time

The successive approximation algorithms of the type

$$c[n] = c[n-1] + \delta c[n-1] \tag{1}$$

— where $c$[n] is the unknown parameter vector in the n-th step and $\delta c$[n − 1] is the calculation increment of the vector in the n-th step, — solve the following task in the stochastic case:

$$P\left\{\lim_{n\to\infty} c[n] = c^*\right\} = 1\,. \tag{2}$$

In the unimodal case $c^*$ is the required value of the parameter vector for an infinite number of the steps and then the algorithm is convergent with a probability of 1, if the conditions of convergence are satisfied.

Let the F[$c$] be a scalar function; then

$$\lim_{n\to\infty} M\{F(c[n] - c^*)\} = \min \tag{3}$$

and with the expected value of the quality criterion in place of the minimum respectively, we obtain:

$$\frac{\partial}{\partial c[n]} \lim_{n\to\infty} M\{F(c[n] - c^*)\} = 0\,. \tag{4}$$

On examining the convergence speed the meaning of the quality criterion F[$c$] mostly involves the variation in the scattering of some quantity, according to the investigations of CYPKIN [1]. Be it in the real-time sampled case

$$T = \text{const., so}$$

$$c[N] = c[t_N] \text{ and } t = NT,$$

where $N = 1, 2 \ldots$ is a series of natural numbers. If it is $t > t_N$, then the form of (2) is:

$$P\{F(c[t_N] - c^*) = \min\} = p_N . \tag{5}$$

By increasing $N$ in the infinite, $p_N$ approaches asymptotically 1. The convergence criterion in the real-time case is:

$$\left| \frac{\partial}{\partial c[t_N]} M\{F(c[t_N] - c^*)\} \right| \leq \varepsilon_N \tag{6}$$

with the error $\varepsilon_N$ being: $\varepsilon_N \geq 0$.

Here the numerical value of $\varepsilon_N$ is greatly problem-dependent: the error limit is determined by the order of magnitude of the signals and the parameters, and the noise.

Let the algorithms $A_1, A_2, \ldots, A_i$ be given and each used frequently in the statistical sense for solving some typical tasks. Then for solving real-time tasks the algorithms for which

$$p_N \leq p_{Ni} \leq 1 \tag{7}$$

holds, are suitable, where $p_{Ni}$ is the expected value of the convergence probability in $N$ steps.

The algorithm which is optimum for solving a typical task is the one, for which (6) is the minimum. In the real-time case we cannot speak of an algorithm having a generally optimum convergence speed, due to the task-dependence but the statistically optimum convergence speed — with the task-dependence but the statistically optimum convergence speed — with the task-dependence disregarded, — may be determined as follows:

Let us denote — for the simulation of $p_N$ of a stochastic convergence probability, — by $\xi_N$ the probability variable equalling 1, when the $N$-th step was optimum with respect to direction and magnitude and equals zero in the opposite case; then the number of the optimum steps is:

$$N_{\text{opt}} = \sum_{N=1}^{N} \xi_N \tag{8}$$

where $N = 1, 2, \ldots N$.

The relative frequency of the optimum steps: $N_{opt}/N$ is also a probability variable. Let us now apply the error finding procedure of the MONTE-CARLO methods to this probability variable [2]; then under very general conditions:

$$N = \frac{9(1-p_N)}{p_N \cdot d^2} \tag{9}$$

where $d = 1 - p_N/p_N$ is the relative error, when $p_N N$ 1, i.e. $N > 1000$ ($> 30$).

We shall look for the error probability of $N_{opt}/N = 1$, so

$$N_{opt} = \frac{9p_N}{1-p_N} \ . \tag{10}$$

Therefore,

| $p_N$ | 0.9 | 0.95 | 0.99 | 0.997 |
|---|---|---|---|---|
| $N$ | 81 | 170 | 891 | ~3000 |

(11)

If the successive steps are independent, — which condition is approximately satisfied in the stochastic case, — then Table 1 contains the minimum number of steps necessary for the required convergence probability. On investigating the number of steps occurring in some typical tasks, we have found that the values given by the table may be used, e.g. in the case of the SARIDIS algorithm [2].

The number of steps is a characteristic, and generally the most important one, of the convergence speed, but finally the calculation time of the operations required in one step must also be taken into account for the realistic evaluation of the convergence speed.

The exact mathematical treatment of the expected number of operations of the algorithmizable tasks is given by FREY in ref. [4].

Let $A_1, A_2 \ldots A_i$ be the algorithms selected for solving a given task. From among these algorithms the one by which the convergence probability is obtained in a minimum time, is regarded as optimum. For solving the task given in [4] the error is proportional with $F(c[N] - c^*)$, and accordingly the estimated scattering of the $i$-th algorithm is:

$$\sigma_i^2 = \frac{1}{N_i - 1} \sum_{N=1}^{N_i} F\big(c[N] - c^*\big)^2 \ . \tag{12}$$

Let $\tau_i$ be the calculation time of the operations required in one step; then by forming the products $(N_i - 1)\tau_i\sigma_i^2$ the algorithm offering the minimum value of this product is optimum for solving the selected task. This implies that the algorithm must be made sensitive for the quantity

$$\sum_{N=1}^{N_i} F(c[N] - c^*) \tag{13}$$

but then $\tau_i' > \tau_i$, if $\tau_i'$ is the calculation time of the operations required in one step with the sensitized algorithm.

On the other hand $\sigma_i^{2\prime} < \sigma_i^2$ and $N_{i\,\mathrm{opt}}' < N_{i\,\mathrm{opt}}$ so the correct solution is:

$$(N_{i\,\mathrm{opt}} - 1)\tau_i > (N_{i\,\mathrm{opt}}' - 1)\tau_i' \,. \tag{14}$$

The expression (14) is suitable for evaluating the convergence acceleration by real-time algorithms, but it says nothing about the error probability $\varepsilon_N$. As the real-time $T_r$ time division is $T_r = \mathrm{const.}$, the accelerated version of the i-th algorithm may be applied, if

$$T_r \geq \mathrm{r}(N_{i\,\mathrm{opt}}' - 1)\tau_i' \tag{14a}$$

is satisfied with a high relative frequency, while the specified convergence probability $p_N$ is constant. When this condition is not satisfied, the usefulness of the result is determined by the evaluation of (13) and (6), respectively. So the evaluation by (14) gives no reliable result.

If the convergence acceleration is characterized by the expression

$$(N_{i\,\mathrm{opt}} - 1)\sigma_i^2\tau_i > (N_{i\,\mathrm{opt}}' - 1)\tau_i'\sigma_i^2 \tag{15}$$

instead of by (14), then this problem is to be avoided.

Let us write the expression (14) in the following form:

$$\sum \varepsilon_{Ni} T_r \geq (N_{i\,\mathrm{opt}}' - 1)\tau_i'\sigma_i^2 = \tau_i' \sum_{N=1}^{N_{i\,\mathrm{opt}}} F(c[n] - c^*)^2 . \tag{15a}$$

The expression (15) defines the accumulated error $\varepsilon_{Ni}$. Be $K_i = \tau_{i0}'/T_r$, which is constant for the given computer and algorithm; with this the new form of (15a) is:

$$\sum \varepsilon_{Ni} \geq \tau_i \sum_{N=1}^{N_i} F(c[n] - c^*)^2 . \tag{16}$$

Now the optimum from among the number $i$ of the algorithms is the one, for which the accumulated error $\varepsilon_{Ni}$ is minimum and in this way the task-dependence may be taken into account. As the required parameter vector is $c$, this cannot be formed recurrently during the calculations, therefore, it is usually substituted by the instantaneous value of the gradient figuring directly, or indirectly in $\delta c[n-1]$. So the steps for selecting an algorithm are:

*a)* the selection of $T_r$,

*b)* the selection of $p_N$, the convergence probability giving $N_{opt}$,

*c)* only algorithms for which

$$\tau_i \leq \frac{T_r}{N_{opt}}$$

applies, may be considered, by which a group of the algorithms was selected with the end of the calculation time of one step,

*d)* the selection of the algorithm having minimum probability of the accumulated error, according to (16), for the actual task.

## Conclusions

For deciding the applicability of the real-time, or evaluating the convergence speed and accuracy of an algorithm, is solved only for specific cases. An algorithm nearly optimal for solving a task can hardly be used for solving other tasks.

As the procedure used for examining error probability in the MONTE-CARLO methods applies to the case of any arbitrary signal distribution, so the described calculation simulating the convergence speed of the successive approximation-type algorithms is generally valid — see table 1, — not regarding the task-dependence.

## Summary

It is difficult to ensure convergence for asymptotically optimum algorithms in case of real time. This paper shows how the convergence rate is possible to be numerically determined in time.

## References

1. CYPKIN, Y.: Grundlage der Theorie Lernender Systeme VEB. Verlag Technik. Berlin. 1972.
2. SARIDIS, G. N.: Learning applied to successive approximation algorithms. IEEE Trans. on System Science and Cyb. Vol. SSC—6. No. 2. Apr. 1970. pp. 97—103.
3. BENCSIK, I.: The expectable number of steps in successive approximation algorithms. Periodica Polytechnica. El. **17**. 1973, 4.
4. FREY, T., Dr. Dissertation, Hungarian Academy of sciences, 1970.

István BENCSIK H-1521 Budapest