Ivan Kramosil

Extremum-searching hierarchical parallel probabilistic algorithms

Kybernetika, Vol. 24 (1988), No. 2, 110--121

Persistent URL: http://dml.cz/dmlcz/125128

KYBERNETIKA — VOLUME *24* (1988), NUMBER 2

EXTREMUM-SEARCHING HIERARCHICAL PARALLEL PROBABILISTIC ALGORITHMS

IVAN KRAMOSIL

Considering a large finite set A together with a mapping f which takes A into a set with linear ordering, the extremum-search problem for $\langle A, f \rangle$ consists in searching for an element of A for which f attains the minimum (or maximum) value. Simple randomization or "non-idealized" parallelism are proved not to improve the situation substantially when compared wit a systematical exhaustive inspection. Therefore, hierarchical parallel probabilistic algorithms for the problem in question are suggested and their time computational complexity is investigated and minimized.

1. INTRODUCTION

Many algorithms of artificial intelligence as well as of other domains of applied mathematics contain sub-algorithms for various extremum-search problems. In its full generality, the extremum-search problem can be described in the following very simple way.

Let $A = \{a_1, a_2, \ldots, a_N\}$ be a non-empty finite set of abstract elements (i.e. their nature is not important for what follows), let B a non-empty set equipped by a linear ordering $\leqq$. Hence, for all $x, y, z \in B$, $x \leqq x$, $x \leqq y \,\&\, y \leqq x \Rightarrow x = y$, $x \leqq y \Rightarrow$ $\Rightarrow (y \leqq z \Rightarrow x \leqq z)$ and $x \leqq y \vee y \leqq x$. Let us mention that throughout all this paper the interpretation of B as the real line $(-\infty, \infty)$ equipped by the usual ordering can be kept in mind as a useful illustration. Let f be a mapping defined on A and taking this set into B, let $A_0(f)$ be the set of elements in A on that f takes its minimum value, i.e.

$$A_0(f) = \{a : a \in A, f(a) \leqq f(b) \text{ for all } b \in A\} = \bigcap_{b \in A} \{a : a \in A, f(a) \leqq f(b)\} . \tag{1}$$

In the sequel, the analogous formulations and solutions for maximum values will be strictly dual and will not be presented explicitly. Extremum (hence, minimum) – search problem for $\langle A, f \rangle$ consists in finding at least one element from $A_0(f)$; as A is finite, $A_0(f)$ is trivially non-empty.

Clearly, our abilities to solve this problem as well as time, space or other demands and expenses connected with a solution, decisively depend on a priori knowledge and testing devices being at our disposal. Let us consider a situation when these tools are rather limited. Namely, take an oracle which contains an element $a \in A$ (this initialization is not substantial, as shall be easily seen) and works as follows: when $b \in A$ is put on the input, the device compares $f(a)$ with $f(b)$ and replaces a by b iff $f(b) \leqq f(a)$; if this is not the case, a is left unchanged. The work finishes by putting at the output the instantaneous value of a, i.e. such an element among the tested ones, for which the value of f is minimum. The device works on the "black-box" principle, so that no insight into the way in which the device decides is possible.

Under the condition that the time necessary to test an element of A is the same for all elements and does not depend on N, and taking into consideration the usual "worst-case" analysis, the time complexity of an exhaustive systematic search for an element from $A_0(f)$ is in the $\mathcal{O}(N)$-class. Clearly, this worst case occurs if $A_0(f)$ is a singleton, its element being tested as the last. As can be easily seen, the most elementary randomization does not improve the situation. Or, applying the Laplace principle and taking the uniform probability distribution over A (each element can be sampled with the same probability N^{-1}), we obtain this probabilistic algorithm: take M statistically independent random samples from the uniform distribution over A and find an element among them with minimal value of f. The probability of error, i.e. the probability that this element is not in $A_0(f)$, can be easily computed to be

$$PE = (1 - (\text{card } A_0)/(\text{card } A))^M . \tag{2}$$

Hence, for card $A_0(f) = 1$, M must be at least $(\ln(1/\varepsilon))\,N$ in order to obtain $PE < \varepsilon$. So, the time complexity is again in $\mathcal{O}(N)$, and this result holds in general, if card $A_0(f)$ is a constant independent of N. However, one advantage of this probabilistic algorithm is worth to mention for what follows: using the systematic exhaustive searching procedure we must assure, somehow, that each element of A is tested just once. When discussing the time complexity of this algorithm, we have not considered the time consumptions necessary to satisfy this demand. In the case of the probabilistic algorithm in question, on the other hand, statistical samples from A have been taken independently and with possible repetitions. All the "past" of the procedure is "hidden" in the instantaneous value $f(a)$ with which the function value for the element just tested is compared. In what follows, we shall try to improve the probabilistic algorithm outlined above by using the fact that random samples and tests of the sampled elements can be taken and done simultaneously by a number of identical copies of the testing oracle (processors). On the other hand, we shall accept rather realistic assumptions concerning the limited possibilities in inspection and cummulation of outputs of parallel processors and the corresponding time demands. These assumptions are realistic particularly in comparison with the usual conventions in the theory of non-deterministic algorithms.

2. TWO-LEVEL EXTREMUM-SEARCHING HIERARCHICAL PARALLEL PROBABILISTIC ALGORITHM

Consider sets A, B, a mapping $f\colon A \rightarrow B$, and $A_0(f)$ as above. The work of the testing oracle described informally in the previous section can be formalized by defining the mapping MIN which ascribes, to the given f and to each finite sequence $\langle a_i \rangle_{i=1}^n$ of elements of A, an element $MIN(f, \langle a_i \rangle_{i=1}^n)$ defined by induction as follows:

$$\text{(3)} \qquad \text{if} \quad n = 1\,. \quad \text{then} \quad MIN(f, \langle a_1 \rangle) = a_1\,,$$

$$MIN(f, \langle a_1, \ldots, a_n, a_{n+1} \rangle) = a_{n+1}\,, \quad \text{if} \quad f(a_{n+1}) \leqq f(MIN(f, \langle a_1, \ldots, a_n \rangle))\,,$$

$$MIN(f, \langle a_1, \ldots, a_n, a_{n+1} \rangle) = MIN(f, \langle a_1, \ldots, a_n \rangle) \quad \text{otherwise}\,.$$

Definition 1. Let A, B, f, $A_0(f)$ be as above, let m, n, k be positive integers. Let $\{X(i,j)\}_{i=1}^{m}\,_{j=1}^{n}$ be an $(m \times n)$-tuple of statistically independent and identically distributed random variables defined on an abstract probability space $\langle \Omega, \mathcal{S}, P \rangle$, taking their values in A and such that, for each $i \leqq m, j \leqq n$ and $a \in A$,

$$\text{(4)} \qquad P(\{\omega\colon \omega \in \Omega,\ X(i,j)(\omega) = a\}) = (\operatorname{card} A)^{-1}\,(=N^{-1})\,.$$

Let $\{Z_l\}_{l=1}^{k}$ be a k-tuple of statistically independent and identically distributed random variables defined on $\langle \Omega, \mathcal{S}, P \rangle$, taking their values in the set $1, 2, \ldots, m$ of integers and such that, for each $l \leqq k$, $s \leqq m$,

$$\text{(5)} \qquad P(\{\omega\colon \omega \in \Omega,\ Z_l(\omega) = s\}) = m^{-1}\,.$$

Then the pair $\mathcal{X} = \langle \{X(i,j)\}_{i=1}^{m}\,_{j=1}^{n}, \{Z_l\}_{l=1}^{k} \rangle$ is called *two-level extremum searching hierarchical parallel probabilistic algorithm* (2-ESHPPA, abbreviately) for the extremum search problem $\langle A, B, f \rangle$. $\mathcal{X}$ can be identified with a random variable taking $\langle \Omega, \mathcal{S}, P \rangle$ into A and defined in this way:

$$\text{(6)} \qquad \mathcal{X}(\omega) = \mathcal{X}(A, B, f, \omega) = MIN(f, \{MIN(f, \{X(Z_l(\omega), j)(\omega)\}_{j=1}^{n})\}_{l=1}^{k})\,.$$

In spite of its rather difficult form, the intuition behind (6) is quite straightforward. Each random variable $X(i,j)$ samples an element from A and the ith processor finds such one among the elements $X(i,1)(\omega), X(i,2)(\omega), \ldots, X(i,n)(\omega)$ for which the value of f is minimum, putting this elements as its own output. Random variables $Z_1, Z_2, \ldots, Z_k$ take random samples from the set of all processors and the (unique) second level processor (supervizor) finds such an element among the output values of the sampled processors, which minimizes the value of f. This value, denoted in (6) by $\mathcal{X}(\omega)$, is called *solution to the extremum search problem* $\langle A, B, f \rangle$ *given by the*

2-ESHPPA $\mathscr{X}$. The solution is *correct*, if $\mathscr{X}(\omega) \in A_0(f)$; it is *wrong* otherwise. *Probability of error* $PE(\mathscr{X}(A, B, f))$ is defined as the probability of wrong solution, i.e.,

$$PE(\mathscr{X}(A, B, f)) = P(\{\omega: \omega \in \Omega,\ \mathscr{X}(A, B, f, \omega) \in A - A_0\})\,, \tag{7}$$

and the value of this probability serves as an important quantitative characteristic of statistical qualities of the 2-ESHPPA in question.

Theorem 1. Let $\mathscr{X} = \langle\{X(i,j)\}_{i=1}^{m}{}_{j=1}^{n}, \{Z_l\}_{l=1}^{k}\rangle$ be a 2-EHSPPA with the parameters m, n, k for the extremum search problem $\langle A, B, f\rangle$, let $\varepsilon > 0$ be given. If $m\,.\,n \geqq (\ln(2/\varepsilon))(\text{card } A/\text{card } A_0(f))$, and $k \geqq \ln(2/\varepsilon)\, m$, then $PE(\mathscr{X}(A, B, f)) < \varepsilon$.

Proof. Random variables $X(i,j)$ take $m\,.\,n$ independent random samples from A. When computing the probability that at least one sample is in $A_0(f)$, we obtain

$$\begin{aligned}
&P(\{\omega: \omega \in \Omega, \sum_{i=1}^{m} \sum_{j=1}^{n} \chi_{A_0(f)}(X(i,j)(\omega)) > 0\}) = \\
&= 1 - P(\bigcap_{i=1}^{m} \bigcap_{j=1}^{n} \{\omega: \omega \in \Omega, X(i,j)(\omega) \in A - A_0(f)\}) = \\
&= 1 - \prod_{i=1}^{m} \prod_{j=1}^{n} P(\{\omega: \omega \in \Omega, X(i,j)(\omega) \in A - A_0(f)\}) = \\
&= 1 - (P(\{\omega: \omega \in \Omega, X(i,j)(\omega) \in A - A_0(f)\}))^{m.n} = \\
&= 1 - (1 - (\text{card } A_0(f)/\text{card } A))^{m.n}
\end{aligned} \tag{8}$$

due to the supposed statistical independence of random variables $X(i,j)$ and their uniform probability distribution over the set A. If $m\,.\,n \geqq (\ln(2/\varepsilon))\,.$
$.\,(\text{card } A/\text{card } A_0(f))$, then

$$\begin{aligned}
&1 - (1 - (\text{card } A_0(f)/\text{card } A))^{m.n} \geqq \\
&\geqq 1 - (1 - (\text{card } A_0(f)/\text{card } A))^{(\ln(2/\varepsilon))(\text{card } A/\text{card } A_0(f))} = \\
&= 1 - ((1 - (\text{card } A_0(f)/\text{card } A)^{\text{card } A})^{(\ln(2/\varepsilon))/\text{card } A_0(f)} \geqq \\
&\geqq 1 - (\mathrm{e}^{-\text{card } A_0(f)})^{(\ln(2/\varepsilon))/\text{card } A_0(f)} = \\
&= 1 - \mathrm{e}^{-\ln(2/\varepsilon)} = 1 - \mathrm{e}^{\ln(\varepsilon/2)} = 1 - (\varepsilon/2)\,.
\end{aligned} \tag{9}$$

Now, if at least one sample $X(i,j)(\omega)$ is in $A_0(f)$, then at least one among the m processors contains a value from $A_0(f)$ as its output value. $\mathscr{X}(\omega) = 1$ iff at least one such output is sampled by some Z_l, $l \leqq k$; due to the assumptions imposed on Z_l we obtain

$$\begin{aligned}
&P(\{\omega: \omega \in \Omega, \mathscr{X}(A, B, f, \omega) \in A_0(f)\}/\{\omega: \omega \in \Omega, \\
&\sum_{i=1}^{m} \sum_{j=1}^{n} \chi_{A_0(f)}(X(i,j)(\omega)) > 0\}) \geqq 1 - (1 - m^{-1})^k \geqq 1 - (1 - m^{-1})^{(\ln(2/\varepsilon)m} \geqq \\
&\geqq 1 - (\mathrm{e}^{-1})^{\ln(2/\varepsilon)} = 1 - (\varepsilon/2)
\end{aligned} \tag{10}$$

analogously as in (9). Combining (9) and (10),

$$(11) \qquad P(\{\omega\colon \omega \in \Omega, \mathscr{X}(A, B, f, \omega) \in A_0(f)\}) =$$

$$= P(\{\omega\colon \omega \in \Omega, \mathscr{X}(A, B, f, \omega) \in A_0(f)\} \mid \{\omega\colon \omega \in \Omega, \sum_{i=1}^{m} \sum_{j=1}^{n} \chi_{A_0(f)}(X(i,j)(\omega)) > 0\}) \,.$$

$$P(\{\omega\colon \omega \in \Omega, \sum_{i=1}^{m} \sum_{j=1}^{n} \chi_{A_0(f)}(X(i,j)(\omega)) > 0\}) \geqq$$

$$\geqq (1 - \varepsilon/2)(1 - \varepsilon/2) > 1 - \varepsilon \,,$$

so that $PE(\mathscr{X}(A, B, f)) < \varepsilon$ and the theorem is proved. □

3. OPTIMIZATION OF COMPUTATIONAL COMPLEXITIES 2-ESHPPA'S

The conditions of Theorem 1 bind the product mn, hence, the natural question arises, which ratio of m and n is the optimal one from the point of view of minimization of the corresponding time computational complexity. To be able to answer the question we are to precise it, i.e. to define quantitative criteria for time computational complexity, namely, the two following most simple criteria originating from [2] will be taken into consideration. Both these criteria will be defined as linear functions of the number of random samples which need to be successively executed, but the time complexity of particular samples will be treated in a different way. If $\mathscr{X}$ is a 2-ESHPPA with parameters m, n and k, then its *unit time (computational) complexity* $TCU(\mathscr{X})$ is defined by $\alpha_1 n + \alpha_2 m + \alpha_3$ with appropriate $\alpha_1, \alpha_2 > 0$, $\alpha_3 \geqq 0$ independent of A, B, and f. The *logarithmic time (computational) complexity* $TCL(\mathscr{X})$ is defined by $\beta_1 n \log_2 \operatorname{card} A + \alpha_2 k \log m + \beta_3$ with appropriate $\beta_1, \beta_2 > 0$, $\beta_3 \geqq 0$ independent of A, B, and f. Hence, in the first case the random sample from A is considered to be of constant time complexity no matter which the sampled element and the cardinality of the sample space may be (but for different set, say, A and $\{1, 2, \ldots, m\}$, the constants may be different). On the other hand, in the logarithmical case the complexity of random samples increases linearly with the binary logarithm of the cardinality of the sample space, as if the samples were realized through a sequence of independent regular coin tosses (so that the length of this sequence must be $\ulcorner \log_2 \operatorname{card} A \urcorner$, with $\ulcorner\,\urcorner$ denoting the upper integer part).

Theorem 2. Consider an extremum search problem $\langle A, B, f\rangle$ and the class $\mathscr{H}$ of all 2-ESHPPA's satisfying the conditions of Theorem 1 with respect to this problem. If $\mathscr{X} \in \mathscr{H}$ with parameters $\tilde{m}_N$, $\tilde{n}_N$, $\tilde{k}_N$ ($N = \operatorname{card} A$) is such that $TCU(\mathscr{X}) \leqq TCU(\mathscr{X}')$ for all $\mathscr{X}' \in \mathscr{H}$, then, for $v = \operatorname{card} A_0(f)$,

$$(12) \qquad \begin{aligned} \tilde{m}_N &= \ulcorner \sqrt{(\alpha_1 \alpha_2^{-1} v^{-1})} \sqrt{N} \urcorner \,, \\ \tilde{n}_N &= \ulcorner \ln (2/\varepsilon) \sqrt{(\alpha_2 \alpha_1^{-1} v^{-1})} \sqrt{N} \urcorner \,, \\ \tilde{k}_N &= \ulcorner \ln (2/\varepsilon)\, \tilde{m}_N \urcorner \,. \end{aligned}$$

If $\mathscr{X} \in \mathscr{H}$ with parameters $\tilde{m}_N$, $\tilde{n}_N$, $\tilde{k}_N$ is such that $TCL(\mathscr{X}) \leqq TCL(\mathscr{X}')$ for all $\mathscr{X}' \in \mathscr{H}$, then

(13)
$$\begin{aligned} \tilde{m}_N &= \lceil \sqrt{(\beta_1 \beta_2^{-1} (\ln 2)\, v^{-1})}\, \sqrt{N} \rceil + f_1(N)\,, \\ \tilde{n}_N &= \lceil \ln(2/\varepsilon) \sqrt{(\beta_2 (\log_2 \mathrm{e})\, \beta_2^{-1} v^{-1})}\, \sqrt{N} \rceil + f_2(N)\,, \\ \tilde{k}_N &= \lceil \ln(2/\varepsilon)\, \tilde{m}_N \rceil + f_3(N)\,, \end{aligned}$$

where $f_i(N)$, $i = 1, 2, 3$, are appropriate $o(\sqrt{N})$-functions.

Proof. For each $x', y', z' \in \langle 0, \infty)$, satisfying $x'y' \geqq v^{-1} \ln(2/\varepsilon) N$, $z' \geqq \ln(2/\varepsilon)\, x'$, and minimizing the value of $\alpha_1 x' + \alpha_2 y' + \alpha_3$ $(= TCU(\mathscr{X}))$ there are x, y, z such that $xy = \ln(2/\varepsilon) N$, $z = \ln(2/\varepsilon)\, x$, and $\alpha_1 x + \alpha_2 y + \alpha_3 = \alpha_1 x' + \alpha_2 y' + \alpha_3$. Putting $y = x^{-1} v^{-1} \ln(2/\varepsilon) N$ and $z = \ln(2/\varepsilon)\, x$ into $\alpha_1 x + \alpha_2 y + \alpha_3$, we obtain

(14)
$$f(x) = \alpha_1 x^{-1} v^{-1} \ln(2/\varepsilon) N + \alpha_2 \ln(2/\varepsilon)\, x + \alpha_3\,.$$

Hence,

(15)
$$(\mathrm{d}/\mathrm{d}x) f(x) = -\alpha_1 v^{-1} \ln(2/\varepsilon) N x^{-2} + \alpha_2 \ln(2/\varepsilon) = 0$$

yields

(16)
$$x^2 = \alpha_1 \alpha_2^{-1} v^{-1} N\,, \quad x = \sqrt{(\alpha_1 \alpha_2^{-1} v^{-1})}\, \sqrt{N}$$

so that, after an easy calculation,

(17)
$$y = (\ln(2/\varepsilon)) \sqrt{(\alpha_2 \alpha_1^{-1} v^{-1})}\, \sqrt{N}\,, \quad z = (\ln(2/\varepsilon))\, x\,,$$

and taking the lowest integer values, we obtain (12).

In the case of the logarithmic criterion we have to minimize the expression

(18)
$$\beta_1 y \log_2 N + \beta_2 z \log_2 x + \beta_3$$

under the same constraints, so setting $y = v^{-1} \ln(2/\varepsilon) N x^{-1}$, $z = \ln(2/\varepsilon)\, x$, and differentiating with respect to x, we need to solve the equation

(19)
$$\begin{aligned} &(\mathrm{d}/\mathrm{d}x)(\beta_1 x^{-1} v^{-1} \ln(2/\varepsilon) N + \beta_2 \ln(2/\varepsilon)\, x \log_2 x + \beta_3) = \\ &= -\beta_1 v^{-1} x^{-2} \ln(2/\varepsilon) N + \beta_2 \ln(2/\varepsilon) \log_2 \mathrm{e}((\mathrm{d}/\mathrm{d}x) x \ln x) = \\ &= -\beta_1 v^{-1} x^{-2} \ln(2/\varepsilon) N + \beta_2 \ln(1/\varepsilon) \log_2 \mathrm{e}(\ln x + 1) = 0\,. \end{aligned}$$

Hence,

(20)
$$\ln x + x^2 = AN/B$$

with

(21)
$$A = -\beta_1 v^{-1} \ln(2/\varepsilon)\,, \quad B = \beta_2 \ln(2/\varepsilon) \log_2 \mathrm{e}\,.$$

Let $\tilde{x}_N$ solve the equation (20), let x_N solve the equation $x^2 = AN/B$, then

(22)
$$\frac{\ln \tilde{x}_N + \tilde{x}_N^2}{x_N^2} = 1\,.$$

Evidently, $\tilde{x}_N < x_N$ for almost all N. Let there be $0 < K < 1$ such that, for infinitely many N's, $\tilde{x}_N \leqq Kx_N$, then

$$(23) \qquad \frac{\ln \tilde{x}_N + \tilde{x}_N^2}{x_N^2} \leqq \frac{\ln Kx_N + K^2 x_N^2}{x_N^2} = K^2 + \frac{\ln K + \ln x_N}{x_N^2} =$$

$$= K^2 + \frac{\ln K + \ln \sqrt{(AB^{-1})} + (1/2) \ln N}{AB^{-1}N} - K^2$$

as $N \to \infty$, hence, for each K', $K < K' < 1$, and for almost all N's, $(\ln \tilde{x}_N + x_N^2)/x_N^2 \leqq$ $\leqq K'$ and this inequality contradicts (22). So $\lim_{N \to \infty} (\tilde{x}_N/x_N) = 1$ and computing the values A and B we obtain (13). The assertion is proved. □

4. MANY-LEVEL HIERARCHIES

Summing the results of Chapters 2 and 3, we can see that two-level hierarchies improve significantly the time complexity for the extremum-searching problem reducing this complexity from the $\mathcal{O}(N)$-class into the $\mathcal{O}(\sqrt{N})$-class (in the case of unit criterion), or from the $\mathcal{O}(N \log N)$ class into the $\mathcal{O}(\sqrt{(N)} \log N)$-class (for logarithmic criterion). As shown in [3], this quadratic speed-up is, in a sense, the optimum which is reachable by two-level hierarchies with independent random choice. On the other hand, the straightforward idea arises to improve the results of foregoing chapters by an appropriate many-level application of the hierarchical principle. Let us begin with a simple example.

Again, consider an extremum-search problem $\langle A, B, f \rangle$ and suppose that $N =$ $= \operatorname{card} A = 2^K$ for a positive integer K (the modifications necessary if it is not the case are not important for what follows). Given $\varepsilon > 0$, set $\delta = \varepsilon/K$, and $N_i =$ $= N/2^i$, $i = 1, 2, \ldots, K$, so that $N_K = 1$. Consider N_1 processors of the first level, each of them taking $\ulcorner 2 \ln (1/\delta) \urcorner$ independent random samples from the uniform probability distribution over A and putting on its output the latest among the sampled elements which minimizes the function f (according to the definition of the operation MIN). Consider N_2 second-level processors, each of them taking, again $\ulcorner 2 \ln (1/\delta) \urcorner$ independent random samples from the uniform probability distribution over the $N_1 = 2N_2$-element set of first-level processors, and putting on its output the last sampled output value, among the sampled first-order processor, which minimizes the function f (again, the operation MIN is applied). Consider N_3 third-level processors and proceed by induction. Finally, there is just one Kth level processor, which takes $\ulcorner 2 \ln (1/\delta) \urcorner$ random samples from the set of the two $(K - 1)$-st level processors with one-half probability of sampling for each of them in each step, and which outputs this one from the (one or two) sampled output values which minimizes the function f. Such a statistical decision procedure can be called *many-level extremum-searching hierarchical parallel probabilistic algorithm* (MLESHPPA, abbreviately) for the problem $\langle A, B, f \rangle$; if this MLESHPPA is denoted by $\mathscr{X}$, its output (result) may be denoted by $\mathscr{X}(\omega)$ and we may ask whether this result is *correct* (i.e. whether $\mathscr{X}(\omega) \in A_0(f)$), or whether it is *wrong* (*false*, i.e. $\mathscr{X}(\omega) \in A - A_0(f)$).

Let us consider the probability $P(\{\omega: \omega \in \Omega,\ X(\omega) \in A_0(f)\})$. Multiplying the number of first-level processors by the number of samples made by each of them we can easily see that at least $N \ln (1/\delta)$ statistically independent random samples from the uniform probability distribution over A have been taken. As easy calculation analogous to that in the proof of Theorem 1 yields that with probability at least $1 - \delta$ at least one processor samples an element from $A_0(f)$ at least once. But, if a processor samples a value from $A_0(f)$, its output value must be also in $A_0(f)$ due to the definition of the MIN-operation. Hence, with a probability at least $1 - \delta$ there is at least one element from $A_0(f)$ among the output values of the N_1 first-level processors. This calculation can be repeated by induction and we obtain: if there is at least one element from $A_0(f)$ among the output values of the ith level processors, then with probability at least $1 - \delta$ there is a value from $A_0(f)$ also among the output values of the $(i + 1)$-st level processors. Combining these conditional probabilities we obtain that the output of the Kth level processor (i.e. $\mathscr{X}(\omega)$) is in $A_0(f)$ with probability at least

$$(1 - \delta)^K > 1 - K\delta = 1 - \varepsilon\,, \tag{24}$$

so that the probability of error is majorized by ε. At each of the K levels we have performed $2 \ln (1/\delta)$ samples, so the number of samples to be taken sequentially is

$$2K \ln (1/\delta) = 2 \log N \ln (K/\varepsilon) = 2 \log N \ln ((\log_2 N)/\varepsilon)\,. \tag{25}$$

Generating straightforwardly the definition of TCU and supposing that $\alpha_i = 1$ for all $i \leqq K$ and $\alpha_{K+1} = 0$, the expression (25) yields immediately $TCU(\mathscr{X})$. In the case of logarithmic criterion with $\beta_i = 1$, $i \leqq K$, and $\beta_{K+1} = 0$, we obtain that

$$\begin{aligned} TCL(\mathscr{X}) &= \sum_{i=0}^{K} \lceil 2 \ln (1/\delta) \rceil \log_2 N_i = \\ &= \sum_{i=0}^{\log_2 N} \lceil 2 \ln ((\log_2 N)/\varepsilon) \rceil \log_2 (N/2^i) = \sum_{i=0}^{\log_2 N} 2i \lceil \ln ((\log_2 N)/\varepsilon) \rceil = \\ &= 2 \lceil \ln ((\log_2 N)/\varepsilon) \rceil (1/2) \lceil \log_2 N \rceil (\lceil \log_2 N \rceil - 1) = \\ &= \lceil \ln ((\log_2 N)/\varepsilon) \rceil ((\log_2 N)^2 - \log_2 N)\,. \end{aligned} \tag{26}$$

Hence, $TCU(\mathscr{X})$ is in $\mathcal{O}(\log N \log \log N)$ and $TCL(\mathscr{X})$ in $\mathcal{O}((\log N)^2 \log \log N)$, so that the result for two-level hierarchies are substantially improved. A result showing the limitations of this improvement will be given later.

Generalizing this example, the formalized definition can read as follows.

Definition 2. *Many-level extremum-searching hierarchical parallel probabilistic algorithm $\mathscr{X}$ for extremum-search problem* $\langle A, B, f \rangle$ is defined by a sequence

$$\mathscr{X} = \langle K, \{N_i\}_{i=1}^{K}, \{k_i\}_{i=1}^{K}, \{X_{ij}\}_{i=1}^{N_1}{}_{j=1}^{k_1}, \{Z(i, j, l)\}_{i=1}^{K-1}{}_{k=1}^{N_{i+1}}{}_{l=1}^{k_{i+1}} \rangle \tag{27}$$

where

$K \geqq 2$, $N_1 > N_2 > \ldots > N_K = 1$, $k_2, k_2, \ldots, k_K$ are positive integers,

$\{X_{ij}\}_{i=1}^{N_1}{}_{j=1}^{k_1}$ is an $(N_1 \times k_1)$-tuple of statistically independent and identically distributed random variables, each of them taking the probability space $\langle \Omega, \mathcal{S}, P \rangle$ into A and such that, for each $a \in A$,

$$P(\{\omega: \omega \in \Omega, X_{ij}(\omega) = a\}) = (\text{card } A)^{-1} \, (= N^{-1}) . \tag{28}$$

For each $i \leqq K - 1$, $k \leqq N_{i+1}$, $l \leqq k_{i+1}$, $Z(i, k, l)$ are statistically independent random variables defined on $\langle \Omega, \mathcal{S}, P \rangle$. Each $Z(i, k, l)$ takes values in $\{1, 2, \ldots, N_i\}$ and for each $j \leqq N_i$, $l \leqq k_{i+1}$,

$$P(\{\omega: \omega \in \Omega, Z(i, k, l)(\omega) = j\}) = N_i^{-1} . \tag{29}$$

$\mathcal{X}$ can be taken as a random variable taking $\langle \Omega, \mathcal{S}, P \rangle$ into A and defined, by induction, as follows.

$$\begin{aligned} Y_k^1(\omega) &= MIN(f, \langle X_{kj}(\omega) \rangle_{j=1}^{k_1}) , \quad k = 1, 2. \ldots, N_1 , \\ Y_k^{i+1}(\omega) &= MIN(f, \langle Y^i_{Z(i,k,j)(\omega)} \rangle_{j=1}^{k_{i+1}} \quad k = 1, 2, \ldots, N_{i+1} , \\ \mathcal{X}(\omega) &= \mathcal{X}(A, B, f, \omega) = Y_1^K(\omega) . \end{aligned} \tag{30}$$

As in the case of 2-ESHPPA's the solution $\mathcal{X}(\omega)$ to the problem $\langle A, B, f \rangle$ is *correct*, if $\mathcal{X}(\omega) \in A_0(f)$, and it is *wrong* (*false*), if $\mathcal{X}(\omega) \in A - A_0(f)$; the probability of error $PE(\mathcal{X}(A, B, f))$ is defined as the probability of wrong solution.

5. OPTIMIZATION OF COMPUTATIONAL COMPLEXITIES FOR HOMOGENEOUS MLESHPPA's

We shall not solve the optimization problem for MLESHPPA's in its full generality and we limit ourselves to certain subclasses of such algorithms.

Definition 3. Let $\varepsilon > 0$, $\lambda > 1$ be given, an MLESHPPA $\mathcal{X}$ with K levels for an extremum search problem $\langle A, B, f \rangle$ is called (ε, λ)-*homogeneous* (is in the class $\mathcal{H}(\varepsilon, \lambda)$), if the following conditions hold

$$K = \ulcorner \log_\lambda \text{card } A \urcorner = \ulcorner \log_\lambda N \urcorner \tag{31}$$

$$N_i / N_{i+1} = \lambda \quad \text{for all} \quad i = 0, 1, 2, \ldots, K - 1 , \quad N_0 = N , \tag{32}$$

$$\begin{aligned} P(\{\omega: \omega \in \Omega, \bigcup_{j=1}^{N_i+1} \{Y_j^{i+1}(\omega)\} \cap A_0(f) \neq \emptyset\} \mid \{\omega: \omega \in \Omega, \bigcup_{j=1}^{N_i} \{Y_j^i(\omega)\} \cap \\ \cap A_0(f) \neq \emptyset\}) \geqq 1 - (\varepsilon / K) \quad \text{for each} \quad i = 1, 2, \ldots, K - 1 . \end{aligned} \tag{33}$$ □

In fact, condition (31) is not necessary, being implied by (32), and is introduced here just because of a more lucidity. Also the condition (33) is rather simple, in spite of its difficult formalization, and it reads: Supposing that there is an element from

$A_0(f)$ on the output of some of the ith level processors, then with the probability at least $1 - (\varepsilon/K)$ there is an element from $A_0(f)$ on the output of some of the $(i + 1)$-st level processors. Combining these conditional probabilities we can easily deduce, using (24), that $PE(\mathscr{X}(A, B, f)) < \varepsilon$ for all $\mathscr{X} \in \mathscr{H}(\varepsilon, \lambda)$.

The optimization problem within the class $\bigcup_{\lambda > 1} \mathscr{H}(\varepsilon, \lambda)$ is solved by the following assertion.

Theorem 3. Given $\varepsilon > 0$, consider the class $\mathscr{H}_\varepsilon = \bigcup_{\lambda > 1} \mathscr{H}(\varepsilon, \lambda)$ of MLESHPPA's for an extremum search problem $\langle A, B, f \rangle$ such that card $A_0(f) = 1$. Let the homogeneous unit time (computational) complexity $TCUH(\mathscr{X})$ of $\mathscr{X}$ be defined by $\sum_{i=1}^{K} k_i$ (generalization of TCU with $\alpha_1 = \alpha_2 = 1$, $\alpha_3 = 0$, to the many-level case). Then λ_N satisfying the equality

$$(34) \qquad (\ln \ln N + \ln \varepsilon^{-1})(\ln \lambda - 1) = (\ln \lambda - 1) \ln \ln \lambda - 1$$

has the property that for each $\mathscr{X} \in \mathscr{H}(\varepsilon, \lambda_N)$, $\mathscr{X}^0 \in \mathscr{H}_\varepsilon$, $TCUH(\mathscr{X}) \leqq TCUH(\mathscr{X}^0)$, hence, the homogeneous unit time complexity is minimized when the ratio between N_i and N_{i+1} is λ_N.

Proof. If $A_0(f)$ is a singleton and if $\{Y_j^i(\omega)\} = A_0(f)$ for some $i \leqq K - 1, j \leqq N_i$, we need $[N_i \ln (1/(\varepsilon/K))]$ independent samples made in total by the $(i + 1)$-st level processors in order to assure the validity of (33). If, moreover, $N_i/N_{i+1} = \lambda$, each processor of the $(i + 1)$-st level must take $[\lambda \ln (1/(\varepsilon/K))]$ samples, hence,

$$(35) \qquad k_i = \ulcorner \lambda \ln (1/(\varepsilon/K)) \urcorner = \ulcorner \lambda \ln (K/\varepsilon) \urcorner = \ulcorner \lambda \ln (\log_\lambda N/\varepsilon) \urcorner ,$$

and, for $\mathscr{X} \in \mathscr{H}(\varepsilon, \lambda)$,

$$(36) \qquad TCUH(\mathscr{X}) = \sum_{i=1}^{K} k_i = K \ulcorner \lambda \ln ((\log_\lambda N)/\varepsilon) \urcorner =$$

$$= \lambda (\log_\lambda N) \ln ((\log_\lambda N)/\varepsilon) = \lambda (\ln N/\ln \lambda) \ln ((\ln N)/((\ln \lambda) \varepsilon)) .$$

The only we need to find is the minimum value of the function

$$(37) \qquad f(x) = (x/\ln x)(\ln \ln N - \ln \ln x - \ln \varepsilon) =$$

$$= (x/\ln x)(\ln \ln N - \ln \varepsilon) - ((x \ln \ln x)/\ln x)_i .$$

But,

$$(38) \qquad (\mathrm{d}/\mathrm{d}x)(x/\ln x) = (\ln x - 1)(\ln x)^{-2} ,$$

$$(39) \qquad (\mathrm{d}/\mathrm{d}x)(x \ln \ln x/\ln x) =$$

$$= (\ln x)^{-2} [(\mathrm{d}/\mathrm{d}x)(x \ln \ln x)] \ln x - x \ln \ln x[(\mathrm{d}/\mathrm{d}x) \ln x]) =$$

$$= (\ln x)^{-2} [\ln x(\ln \ln x + (\ln x)^{-1}) - \ln \ln x] ,$$

so that

$$(40) \qquad (\mathrm{d}f(x)/\mathrm{d}x) = (\ln x)^{-2} (\ln \ln N + \ln \varepsilon^{-1}) (\ln x - 1) - \\ - (\ln x)^{-2} ((\ln x - 1) \ln \ln x + 1),$$

and this value is zero just when (34) holds, so the assertion is proved. □

The only value λ_N which minimizes $TCUH(\mathscr{X})$, and which is in $o(\ln N)$-class, can be approximated by $\lambda_N \approx \mathrm{e} = 2{\cdot}718 \ldots$ in the following sense. Take the function

$$(41) \qquad h(\lambda, N) = (\lambda/\ln \lambda) \ln N \ln \ln N,$$

if $\mathscr{X} \in \mathscr{H}(\varepsilon, \lambda)$, then

$$(42) \qquad \lim_{N\to\infty} \frac{TCUH(\mathscr{X})}{h(\lambda, N)} = \lim_{N\to\infty} \frac{(\lambda/\ln \lambda) \ln N(\ln \ln N - \ln \ln \lambda_N \varepsilon)}{(\lambda/\ln \lambda) \ln N \ln \ln N} = \\ = 1 - \lim_{N\to\infty} \frac{\ln \ln \lambda_N \varepsilon}{\ln \ln N} \geqq 1 - \lim_{N\to\infty} \frac{\ln \ln \ln N}{\ln \ln N} = 1.$$

Hence, $TCUH(\mathscr{X}) \approx h(\lambda, N)$ in the sense of (42) and $h(\lambda, N)$, taken as a function of λ, is minimized by $\lambda = \mathrm{e}$ (as $(\mathrm{d}/\mathrm{d}\lambda)(\lambda/\ln \lambda) = (\ln \lambda)^{-2} (\ln \lambda - 1)$). The fact that only $\lambda_N \in o(\ln N)$ are considered is not restrictive, as for $\lambda_N \in \mathcal{O}(\ln N)$ we obtain

$$(43) \qquad TCUH(\mathscr{X}) \in \mathcal{O}((\ln N)^2)$$

and this result is qualitatively worse than that with $\lambda = const$ (when we were in $\mathcal{O}(\ln N \ln \ln N)$). For the same reasons also λ_N increasing more rapidly than $\ln N$ can be avoided from our considerations.

Let us close this paper by some remarks concerning thr further possibilities of investigation and development in the domain in question. Some of them are inspired by [4], when we have investigated similar hierarchical parallel algorithmic structures for the searching problem; this problem consists in answering the question whether a subset of the basic set A is empty or not or not supposing that the only device being at hand is an oracle which tests whether at random sampled elements of V are in the subset in question or not. So the following problems may be considered:

- optimization of $\mathscr{X}$ for broader classes than those investigated here.
- instead of probability of error in the sense defined above (i.e. $\mathscr{X}(\omega) \in A - A_0(f)$) some other criterion should be optimized, e.g., the expected distance $\mathsf{E}\varrho(\mathscr{X}(\omega), A_0(f))$ between $\mathscr{X}(\omega)$ and $A_0(f)$ should be minimized supposing that A is equipped by a metric ϱ and $\varrho(\mathscr{X}(\omega), A_0(f)) = \min_{x \in A_0(f)} \varrho(\mathscr{X}(\omega), x)$.
- instead of "classical" worst case analysis the Bayesian one can be applied, so that the extremum-search problem $\langle A, B, f \rangle$ is supposed to be sampled at random (by the so-called apriori distribution) and probabilities of error or other qualitative characteristics for the worst cases are replaced by their expected values with

respect to the apriori distribution. This involves a number of problems connected with the Bayesian approach in general, and we do not go here into details.

- the possibility of failure of the oracle which compares $f(a)$ and $f(b)$, for $a, b \in A$ can be incorporated into the model, i.e. we may suppose that if $f(a) \leqq f(b)$, then the device gives this correct answer with probability $1 - \varepsilon < 1$, and gives the wrong answer $f(a) > f(b)$ with positive probability ε. In [4] we have investigated this possibility of failure for the searching problem, and the way in which the probability of failure combines with the other uncertainties within our model proved to lead to qualitatively different results than those in the case of failure-proof oracles.

In the list below, Feller's monography may serve for all references concerning the elementary notions, models and results of probability theory used in this paper.

(Received April 30, 1987.)

REFERENCES

[1] W. Feller: An Introduction to Probability Theory and its Applications, Vol. I. Second edition. John Wiley and Sons — Chapman and Hall, New York—London 1957 (Russian translation: Mir, Moscow 1964).

[2] U. Manber and M. Tompa: The complexity of problems on probabilistic, non-deterministic, and alternating decision trees. J. Assoc. Comput. Mach. *32* (1985), 3, 720—732.

[3] J. Reif: On synchronous parallel computations with independent probabilistic choice. SIAM J. Comput. *13* (1984), 1, 46—55.

[4] I. Kramosil: Hierarchical connection of probabilistic approach and parallelism in searching tasks of artificial intelligence (in Czech). In: Aplikace umělé inteligence AI' 87, 23—31.

RNDr. Ivan Kramosil, DrSc., Ústav teorie informace a automatizace ČSAV (Institute of Information Theory and Automation — Czechoslovak Academy of Sciences). Pod vodárenskou věží 4, 182 08 Praha 8. Czechoslovakia.