

# МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ, КОМПЛЕКСОВ И КОМПЬЮТЕРНЫХ СЕТЕЙ (ТЕХНИЧЕСКИЕ НАУКИ) (2.3.5)

УДК 004.932:004.921:004.627

DOI: 10.24160/1993-6982-2023-1-136-144

## О поисковом синтезе предиктора для сжатия изображений

П.А. Чернов

В области сжатия полутоновых изображений без потерь высокие результаты сжатия обеспечиваются предиктивной схемой сжатия. Одним из предикторов является предиктор *GAP* (*Gradient Adjusted Prediction*). В данной статье исследуется возможность поискового синтеза предиктора, наподобие предиктора *GAP*, рассмотрена возможность поискового синтеза оптимальных модельных параметров предиктора для каждого возможного контекста предиктора *GAP*, при этом исходные изображения разбивались на контексты с множествами окрестных точек.

При реализации поискового синтеза предиктора, наподобие предиктора *GAP*, использован классический генетический алгоритм, однако с целью улучшения работы классического генетического алгоритма было проведено исследование генераторов псевдослучайных чисел с равномерным распределением, построенных по методу мультиплексивного датчика. Синтез генераторов псевдослучайных чисел с равномерным законом распределения был автоматизирован, автоматически выполняется проверка статистической гипотезы о типе распределения обеспечиваемого генератором псевдослучайных чисел, для этого вычисляется значение распределения Хи-квадрат, выполняется упрощенный расчет Гамма-функции и численное интегрирование по формуле Симпсона. В итоге были использованы генераторы псевдослучайных чисел не встроенные в инструментальные средства программирования, а подобранные самостоятельно, периоды подобранных генераторов псевдослучайных чисел находятся в диапазоне от 402 544 072 до 2 544 344 962.

Дискретный характер определения контекстов *GAP* в оригинальном *GAP* приводит к заметному завышению энтропии ошибки предсказания. Так, для контекста 32 разница энтропии, достигнутой в результате оптимизации модельных параметров, и энтропии достигнутой оригинальным *GAP* предиктором, составляет 0,1148 *bpp*, для контекста 33 — 0,02568 *bpp*, для контекста 8 — 0,07011 *bpp*, для контекста 9 — 0,00925 *bpp*. Предиктор *GAP* имеет не много возможностей для улучшения предсказания за счет определения оптимальных параметров работы предиктора для каждого возможного контекста. Однако по причине малой вычислительной сложности предиктора *GAP* важно исследовать возможности по улучшению эффективности предиктора *GAP*.

**Ключевые слова:** предиктор, сжатие изображений без потерь, *GAP*, поисковый синтез, генетический алгоритм.

**Для цитирования:** Чернов П.А. О поисковом синтезе предиктора для сжатия изображений // Вестник МЭИ. 2023. № 1. С. 136—144. DOI: 10.24160/1993-6982-2023-1-136-144.

## About Search Synthesis of the Predictor for Image Compression

P.A. Chernov

In the field of lossless halftone image compression, high compression results are provided by a predictive compression scheme. One predictor is the *GAP* (*Gradient Adjusted Prediction*) predictor. This article explores the possibility of predictor search synthesis, like *GAP* predictor, considers the possibility of search synthesis of optimal predictor model parameters for each possible *GAP* predictor context, while the test images were divided into contexts with sets of neighborhood pixels.

When implementing the search synthesis of a predictor, like the *GAP* predictor, a classical genetic algorithm was used, however, in order to improve the operation of the classical genetic algorithm, a study was carried out on pseudo-random number generators with a

uniform distribution, built using the multiplicative sensor method. The synthesis of pseudo-random number generators with a uniform distribution was automated, the statistical hypothesis on the type of distribution provided by the pseudo-random number generator is automatically tested, for this the value of the Chi-square distribution is calculated, a simplified calculation of the Gamma function and numerical integration according to the Simpson formula is performed. As a result, pseudo-random number generators were not built into the programming tools, but selected on their own, the periods of selected pseudo-random number generators are in the range from 402 544 072 to 2 544 344 962.

The discrete nature of defining *GAP* contexts in the original *GAP* leads to a noticeable overestimation of the entropy of the prediction error. Thus, for context 32, the difference between entropy achieved as a result of optimization of model parameters and entropy achieved by the original *GAP* predictor is 0.1148 *bpp*, for context 33 — 0.02568 *bpp*, for context 8 — 0.07011 *bpp*, for context 9 — 0.00925 *bpp*. The *GAP* predictor does not have many possibilities to improve prediction by determining optimal predictor performance parameters for each possible context. However, due to the low computational complexity of the *GAP* predictor, it is important to investigate opportunities to improve the effectiveness of the *GAP* predictor.

*Key words:* predictor, lossless image compression, *GAP*, search synthesis, genetic algorithm.

*For citation:* Chernov P.A. About Search Synthesis of the Predictor for Image Compression. Bulletin of MPEI. 2023;1:136—144. (in Russian). DOI: 10.24160/1993-6982-2023-1-136-144.

## Введение

Методы сжатия изображений иногда [1] делят на практические алгоритмы и алгоритмы определяющие теоретические пределы степени сжатия часто сопровождающиеся очень низкими скоростями кодирования, декодирования. Один из подходов позволяющих создать высокоеффективный практический метод — это предиктивная схема. Предиктор оперирует множеством точек из окрестности предсказываемой точки, на базе которых строит предсказание.

«Предиктора причинная окрестность» (ППО) — множество точек изображения, на основе которых предсказывается значение очередной кодируемой точки.

В данной статье рассматривается синтез предиктора наподобие предиктора *GAP*. Предиктор *GAP* реализуется следующим образом. ППО *GAP* изображена на рис. 1, текущая кодируемая точка, значение в которой предсказывается, обозначена *y*.

Сначала вычисляются вертикальный  $d_v$  и горизонтальный  $d_h$  градиенты:

$$d_h = |w - ww| + |n - nw| + |ne - nne|;$$

$$d_v = |w - nw| + |n - nn| + |ne - nne|.$$

Затем предсказание выполняется в соответствии с псевдокодом, записанным так:

```
if ( $d_v - d_h > 80$ ) predGAP = w
else if ( $d_v - d_h < -80$ ) predGAP = n
else {
    predGAP = (w + n) / 2 + (ne - nne) / 4
    if ( $d_v - d_h > 32$ ) predGAP = (predGAP + w) / 2
    else if ( $d_v - d_h > 8$ ) predGAP = (3 * predGAP + w) / 4
    else if ( $d_v - d_h < -32$ ) predGAP = (predGAP + n) / 2
    else if ( $d_v - d_h < -8$ ) predGAP = (3 * predGAP + n) / 4
}
```

В [2] была предпринята попытка определить оптимальные параметры работы *GAP* предиктора, но не была предпринята попытка подобрать оптимальные значения параметров предиктора для каждого возможного контекста *GAP*-предиктора, определяемого различий вертикального и горизонтального градиентов.

	$X_6$	$X_9$
	<i>nn</i>	<i>nne</i>
$X_3$	$X_2$	$X_4$
<i>nw</i>	<i>n</i>	<i>ne</i>

$X_5$	$X_1$	<i>y</i>
<i>ww</i>	<i>w</i>	

Рис. 1. ППО *GAP*

## Генератор псевдослучайных чисел по методу мультипликативного датчика

Проследовав по методике из [3], построю генераторы псевдослучайных чисел с равномерным распределением по методу мультипликативного датчика.

В генераторе псевдослучайных чисел с равномерным распределением по методу мультипликативного датчика очередное псевдослучайное число  $x_{i+1}$  получим по формуле

$$x_{i+1} = (a \cdot x_i + b) \bmod M,$$

где  $a$ ,  $b$ ,  $M$  — натуральные параметры генератора;  $x_i$  — предыдущее натуральное псевдослучайное число. Пусть  $M$  и  $a$  равны простым числам, причем значение параметра  $a$  имеет порядок корня квадратного из  $M$ . Простые числа подберем по формуле Эйлера  $y = x^2 + x + 41$ .

Разбив динамический диапазон выходных значений генератора псевдослучайных чисел на некоторое число интервалов, можно получить гистограмму. Чем ближе теоретическая и экспериментальная гистограммы, тем распределение плотности вероятности случайных чисел экспериментальной последовательности ближе к теоретическому распределению. В качестве метрики близости гистограмм теоретического и экспериментального распределений применим величину  $\chi^2$ , определяемую как

$$\chi^2 = \sum_{j=0}^{L-1} \frac{(v_j - w_j)^2}{w_j},$$

где  $w_j$  — теоретическое число псевдослучайных чисел, попадающих в  $j$ -й столбец гистограммы;  $v_j$  — экспериментальное число псевдослучайных чисел, попадающих в  $j$ -й столбец гистограммы;  $L$  — число столбцов в гистограмме.

Распределение Хи-квадрат задается формулой [4, с. 146]:

$$F_f(x) = P\left\{\chi^2 \leq x\right\} = \frac{\int_0^x t^{\frac{n}{2}-1} e^{-\frac{t}{2}} dt}{2^{\frac{n}{2}} \Gamma\left(\frac{n}{2}\right)},$$

где  $n$  — число степеней свободы,  $n = L - 1$ ;  $\Gamma$  — гамма-функция (расширение понятия факториала на множество вещественных и комплексных чисел);  $\chi^2$  — значение  $\chi^2$ ;  $x$  — квантиль.

Примем следующие ограничения на число степеней свободы  $n$ :

$$n = 2j: j \in N, 5 \leq j \leq 25,$$

где  $N$  — множество натуральных чисел. Тогда вычисление Гамма-функции  $\Gamma$  можно выполнить по упрощенной формуле

$$\Gamma(z) = (z-1)!.$$

Для определения числителя в формуле распределения Хи-квадрат выполнил численное интегрирование по формуле Симпсона [5, с. 380]

$$I \approx \frac{h}{6} \left( f_0 + f_k + 4 \sum_{i=1}^k f_{i-0.5} + 2 \sum_{i=1}^{k-1} f_i \right);$$

$$f_{i-0.5} = f(x_{i-0.5}); \quad x_{i-0.5} = \frac{x_{i-1} + x_i}{2},$$

где  $h$  — шаг;  $x_{i-0.5}$  — середина элементарного отрезка;  $k$  — число элементарных отрезков интегрирования.

При реализации классического генетического алгоритма использованы следующие генераторы псев-

дослучайных чисел с равномерным распределением, построенные по методу мультипликативного датчика:

•  $a = 20063, x_0 = 26283, M = 402544073$ , период 402544072;

•  $a = 20347, x_0 = 14530, M = 414020797$ , период 414020796;

•  $a = 33347, x_0 = 26208, M = 1112055797$ , период 1112055796;

•  $a = 50441, x_0 = 5706, M = 2544344963$ , период 2544344962.

### Исследование предиктора *GAP*

Описываемая методика исследования предиктора *GAP* реализована в [6]. Предположим, что низкочастотная составляющая предиктора *GAP GAPLFMode* может принимать одно из значений:  $(w+n)/2 + (ne-nw)/4; (w+n)/2; (w+n+nw)/3; (w+n+nw+nn)/4, (w+n+n+nw+ww)/4; (w+n+nw+nn+ww)/5$ . Задачу определения лучшего способа определения низкочастотной составляющей предиктора решал методом полного перебора. После того, как найдены значения  $w_0, w_1$  в ходе оптимизации классическим генетическим алгоритмом (обширный обзор методов оптимизации дан в [7]), остается перебрать при фиксированных  $w_0, w_1$  значения низкочастотной части предиктора *GAP GAPLFMode*.

ДМРОЗ [8] выглядит согласно рис. 2, ПП — полный перебор,  $SC_{2,1} = SC_{2,3} = SC_{2,2^{*N}-1}$  — достижение числа генераций, равного 3000,  $SC_{1,1} = SC_{2,2} = SC_{2,4} = SC_{2,2^{*N}}$  — однократное повторение;  $pozl_{2,1} = pozl_{2,3} = pozl_{2,2^{*N}-1} = = \{w_0[i], w_1[i]\}, i = 1, \dots, N, N = 3071, pozl_{2,2} = pozl_{2,4} = pozl_{2,2^{*N}} = \{GAPLFMode\}$ . Поскольку наиболее часто встречаются контексты *GAP* с разностью вертикального и горизонтального градиента не более 80 по модулю, то приведу (табл. 1, 2) оптимальные веса для каждого контекста, разность градиентов для которого не более 80 по модулю.

В частном случае (с фиксированной низкочастотной составляющей) код предиктора может быть записан приблизительно следующим образом:

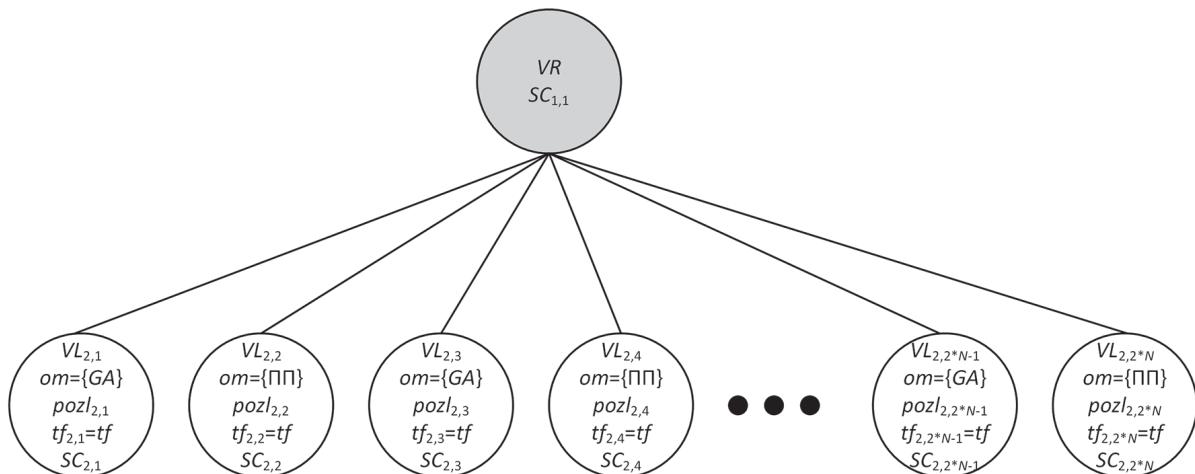


Рис. 2. ДМРОЗ поискового синтеза предиктора *GAP*

Таблица 1

Первая половина *GAP*-контекстов

<i>GAP</i> -контекст	Число <i>nGAPVicinities</i>	<i>GAPLfMode</i>	<i>w</i> <sub>0</sub>	<i>w</i> <sub>1</sub>	<i>SGAPCR</i>	<i>SGAEntropy</i>	<i>GAPCR</i>	<i>GAPEntropy</i>
1	17154	1	0,787982	0,212018	2,10333	3,8035	2,09375	3,82089
2	14676	1	0,666667	0,333333	2,04984	3,90274	2,03376	3,93359
3	11842	1	0,782989	0,217011	1,96114	4,07926	1,94931	4,10402
4	9344	1	0,837743	0,162257	1,862	4,29646	1,85165	4,32047
5	7456	1	0,7504	0,2496	1,77801	4,49941	1,76355	4,53631
6	5985	1	0,730963	0,269037	1,71131	4,67479	1,69636	4,71598
7	4981	1	0,626284	0,373716	1,63729	4,88612	1,61666	4,94847
8	4285	1	0,653304	0,346696	1,60717	4,9777	1,58485	5,04781
9	3833	1	0,606994	0,393006	1,55788	5,13519	1,55508	5,14444
10	3367	1	0,676706	0,323294	1,52864	5,23341	1,52307	5,25255
11	3097	1	0,488091	0,511909	1,52544	5,2444	1,51818	5,26948
12	2853	1	0,565454	0,434546	1,51186	5,29148	1,50324	5,32183
13	2730	1	0,642893	0,357107	1,48071	5,40281	1,47627	5,41906
14	2536	1	0,601717	0,398283	1,49177	5,36274	1,48451	5,38899
15	2348	1	0,522263	0,477737	1,47978	5,40621	1,47072	5,43952
16	2261	1	0,585264	0,414736	1,47704	5,41622	1,46649	5,45521
17	2050	1	0,519534	0,480466	1,4653	5,45963	1,45589	5,49492
18	2095	1	0,491851	0,508149	1,46488	5,46119	1,45171	5,51075
19	1846	1	0,469673	0,530327	1,46089	5,47611	1,4507	5,51459
20	1716	1	0,475917	0,524083	1,44965	5,51859	1,43921	5,5586
21	1690	1	0,343162	0,656838	1,46147	5,47394	1,43933	5,55812
22	1580	1	0,5	0,5	1,44591	5,53285	1,43693	5,56743
23	1585	1	0,438486	0,561514	1,44555	5,53423	1,43228	5,58551
24	1389	1	0,366002	0,633998	1,44045	5,55384	1,42158	5,62754
25	1358	1	0,473126	0,526874	1,45716	5,49014	1,44228	5,54677
26	1306	1	0,376704	0,623296	1,42615	5,6095	1,41299	5,66176
27	1257	1	0,45443	0,54557	1,41633	5,64841	1,40803	5,68172
28	1182	1	0,419077	0,580923	1,43593	5,57129	1,42064	5,63128
29	1157	1	0,513514	0,486486	1,42082	5,63057	1,40843	5,68009
30	1045	1	0,380092	0,619908	1,4369	5,56752	1,42121	5,62899
31	1063	1	0,47741	0,52259	1,41407	5,65745	1,39041	5,75371
32	992	1	0,404226	0,595774	1,43603	5,5709	1,40704	5,6857
33	931	1	0,478908	0,521092	1,40876	5,67874	1,40242	5,70442
34	851	1	0,338026	0,661974	1,40422	5,69712	1,39222	5,74624
35	838	2	0,526954	0,473046	1,41199	5,66577	1,40375	5,69901
36	806	1	0,291667	0,708333	1,40844	5,68003	1,39972	5,71542
37	743	1	0,282309	0,717691	1,39409	5,73851	1,39175	5,74814
38	754	1	0,396406	0,603594	1,38913	5,75901	1,37789	5,80597
39	708	1	0,494378	0,505622	1,41878	5,63863	1,41384	5,65836
40	653	1	0,431312	0,568688	1,39586	5,73125	1,38772	5,76485
41	653	1	0,353977	0,646023	1,3999	5,7147	1,39074	5,75234
42	628	1	0,4111	0,5889	1,42525	5,61305	1,41433	5,6564

Продолжение табл. 1. на стр. 140

Окончание табл. 1

<i>GAP-контекст</i>	Число <i>nGAPVicinities</i>	<i>GAPLfMode</i>	<i>w<sub>0</sub></i>	<i>w<sub>1</sub></i>	<i>SGAPCR</i>	<i>SGAPEntropy</i>	<i>GAPCR</i>	<i>GAPEntropy</i>
43	553	1	0,371429	0,628571	1,40443	5,69625	1,38635	5,77055
44	517	1	0,452513	0,547487	1,41475	5,65473	1,40804	5,68165
45	546	1	0,350185	0,649815	1,39268	5,74431	1,38217	5,788
46	491	1	0,261968	0,738032	1,40558	5,69162	1,39255	5,74485
47	453	1	0,423717	0,576283	1,39742	5,72485	1,38614	5,77144
48	453	1	0,39243	0,60757	1,42364	5,61939	1,40956	5,67552
49	471	1	0,297947	0,702053	1,38666	5,76927	1,37876	5,80233
50	423	1	0,212457	0,787543	1,40255	5,7039	1,38495	5,7764
51	393	1	0,308518	0,691482	1,40846	5,67998	1,38915	5,75891
52	417	1	0,215628	0,784372	1,41511	5,65327	1,39333	5,74164
53	375	1	0,264417	0,735583	1,40362	5,69953	1,36948	5,84165
54	363	1	0,125	0,875	1,41306	5,66147	1,38883	5,76024
55	381	1	0,277632	0,722368	1,45823	5,48611	1,4321	5,58618
56	361	1	0,109885	0,890115	1,43582	5,57173	1,38585	5,77265
57	331	1	0,386265	0,613735	1,4315	5,58856	1,40828	5,68068
58	340	1	0,185144	0,814856	1,45542	5,49668	1,41546	5,65187
59	344	1	0,239765	0,760235	1,42643	5,6084	1,38746	5,76594
60	313	1	0,17191	0,82809	1,41411	5,65729	1,37895	5,8015
61	288	1	0,19743	0,80257	1,4552	5,49754	1,40525	5,69292
62	292	1	0,245073	0,754927	1,4437	5,54133	1,39736	5,72507
63	241	1	0,287902	0,712098	1,47829	5,41166	1,45465	5,49959
64	236	1	0,179012	0,820988	1,49507	5,35091	1,43932	5,5582
65	221	1	0,0585953	0,941405	1,4249	5,61442	1,37088	5,83567
66	236	1	0,254537	0,745463	1,45645	5,49282	1,42388	5,61845
67	243	1	0,0219425	0,978057	1,45018	5,51654	1,38953	5,75735
68	224	1	0,126742	0,873258	1,47495	5,4239	1,41439	5,65615
69	203	1	0,338832	0,661168	1,46806	5,44937	1,41718	5,64502
70	201	1	0,129724	0,870276	1,51347	5,28585	1,4357	5,5722
71	181	3	0,0750973	0,924903	1,54487	5,17843	1,44345	5,54226
72	202	1	0,0163851	0,983615	1,47277	5,43194	1,41192	5,66605
73	185	1	0,152552	0,847448	1,48048	5,40364	1,40437	5,6965
74	169	1	0,190339	0,809661	1,5184	5,2687	1,4626	5,46973
75	181	5	0,0172089	0,982791	1,47324	5,43022	1,40496	5,6941
76	152	1	0,153846	0,846154	1,48942	5,3712	1,43002	5,59432
77	152	1	0,0319183	0,968082	1,5618	5,12229	1,45254	5,50758
78	146	1	0,125143	0,874857	1,55612	5,14098	1,48361	5,39224
79	137	1	0,107522	0,892478	1,52772	5,23655	1,4833	5,39337
80	142	1	0,0583934	0,941607	1,53448	5,21351	1,41987	5,63432

Таблица 2

Вторая половина *GAP*-контекстов

<i>GAP</i> -контекст	Число <i>nGAPVicinities</i>	<i>GAPLfMode</i>	<i>w</i> <sub>0</sub>	<i>w</i> <sub>1</sub>	<i>SGAPCR</i>	<i>SGAPEntropy</i>	<i>GAPCR</i>	<i>GAPEntropy</i>
1536	18161	1	0,949863	0,0501373	2,13381	3,74917	2,13305	3,7505
1537	17086	2	0,990766	0,00923376	2,11678	3,77932	2,1103	3,79093
1537	17086	1	0,990766	0,00923376	2,11177	3,78830	2,1103	3,79093
1538	15035	1	0,966927	0,0330734	2,0496	3,9032	2,04895	3,90444
1539	12453	1	1	0	1,97105	4,05876	1,97105	4,05876
1540	10024	1	1	0	1,89746	4,21615	1,89746	4,21615
1541	8080	1	1	0	1,81901	4,39799	1,81901	4,39799
1542	6618	1	1	0	1,74434	4,58626	1,74434	4,58626
1543	5236	1	0,990377	0,00962316	1,67551	4,77467	1,67443	4,77776
1544	4553	1	0,901278	0,0987219	1,62928	4,91014	1,62724	4,91629
1545	3980	1	0,960709	0,0392911	1,56451	5,11343	1,5621	5,12131
1546	3449	1	0,961012	0,0389881	1,55277	5,15208	1,54622	5,17391
1547	3230	1	0,995393	0,00460693	1,53367	5,21625	1,5264	5,24109
1548	2805	1	0,945702	0,0542981	1,52333	5,25164	1,51576	5,27789
1549	2676	1	0,979224	0,0207759	1,50642	5,31061	1,50265	5,32392
1550	2488	1	0,892939	0,107061	1,48331	5,39336	1,48016	5,40482
1551	2301	1	0,811559	0,188441	1,45984	5,48006	1,45677	5,49159
1552	2142	1	0,909437	0,0905629	1,46707	5,45306	1,46489	5,46115
1553	2027	1	0,820303	0,179697	1,45326	5,50488	1,45137	5,51202
1554	1866	1	0,818273	0,181727	1,45576	5,49541	1,44922	5,52022
1555	1793	1	0,836442	0,163558	1,43829	5,56218	1,43622	5,57019
1556	1660	1	0,855739	0,144261	1,43041	5,59279	1,42751	5,60418
1557	1636	1	0,813997	0,186003	1,42148	5,62795	1,41981	5,63454
1558	1452	1	0,650567	0,349433	1,432	5,58658	1,43079	5,59132
1559	1422	1	0,86778	0,13222	1,41635	5,64834	1,41231	5,66449
1560	1384	1	0,865653	0,134347	1,42499	5,61407	1,42176	5,62681
1561	1306	1	0,694162	0,305838	1,41115	5,66914	1,40949	5,67582
1562	1206	1	0,731107	0,268893	1,41916	5,63715	1,41792	5,64208
1563	1245	1	0,63778	0,36222	1,40427	5,69692	1,39664	5,72802
1564	1206	1	0,915241	0,0847594	1,43135	5,58912	1,42638	5,6086
1565	1130	1	0,416295	0,583705	1,41109	5,66937	1,40531	5,69271
1566	1067	1	0,563006	0,436994	1,39522	5,73387	1,39292	5,74331
1567	980	1	0,644205	0,355795	1,40745	5,68404	1,40466	5,69531
1568	1015	1	0,492469	0,507531	1,43057	5,59219	1,42712	5,60571
1569	935	1	0,704616	0,295384	1,41443	5,656	1,40887	5,6783
1570	873	1	0,442004	0,557996	1,40899	5,67784	1,4045	5,69599
1571	877	1	0,456559	0,543441	1,38314	5,78393	1,37533	5,8168
1572	803	1	0,765327	0,234673	1,41923	5,63686	1,41004	5,67362
1573	817	1	0,285875	0,714125	1,42749	5,60425	1,41809	5,6414
1574	779	1	0,605627	0,394373	1,41434	5,65634	1,40081	5,711
1575	728	1	0,567574	0,432426	1,42049	5,63185	1,40802	5,68173
1576	741	1	0,471713	0,528287	1,39116	5,75061	1,38908	5,75921

Продолжение табл. 2. на стр. 142

Окончание табл. 2

<i>GAP-контекст</i>	Число <i>nGAPVicinities</i>	<i>GAPLfMode</i>	<i>w<sub>0</sub></i>	<i>w<sub>1</sub></i>	<i>SGAPCR</i>	<i>SGAPEntropy</i>	<i>GAPCR</i>	<i>GAPEntropy</i>
1577	704	1	0,44162	0,55838	1,37359	5,82417	1,36511	5,86032
1578	688	1	0,754275	0,245725	1,40883	5,67847	1,40204	5,70596
1579	658	1	0,426462	0,573538	1,40104	5,71005	1,39238	5,74555
1580	603	1	0,585264	0,414736	1,39625	5,72962	1,38977	5,75634
1581	605	1	0,388543	0,611457	1,42244	5,62415	1,41671	5,64688
1582	656	1	0,474792	0,525208	1,37836	5,80399	1,37119	5,83437
1583	547	1	0,297297	0,702703	1,44202	5,54779	1,42981	5,59514
1584	528	1	0,351899	0,648101	1,40508	5,69364	1,39399	5,73892
1585	485	1	0,204082	0,795918	1,39214	5,74655	1,37699	5,80978
1586	495	1	0,118411	0,881589	1,38646	5,77011	1,36607	5,85622
1587	514	1	0,304853	0,695147	1,3998	5,7151	1,39175	5,74816
1588	482	1	0,0414379	0,958562	1,39541	5,73307	1,37539	5,81655
1589	493	1	0,147201	0,852799	1,38971	5,75658	1,37624	5,81296
1590	432	1	0,320916	0,679084	1,40842	5,68013	1,39737	5,72503
1591	457	1	0,472401	0,527599	1,37814	5,80492	1,3718	5,83177
1592	431	1	0,407312	0,592688	1,41205	5,66552	1,39801	5,72243
1593	384	1	0,393898	0,606102	1,41344	5,65995	1,38647	5,77003
1594	398	1	0,357156	0,642844	1,38283	5,78522	1,36955	5,84134
1595	363	1	0,291667	0,708333	1,39426	5,7378	1,36515	5,86015
1596	350	1	0,419973	0,580027	1,37934	5,79987	1,36173	5,87486
1597	357	1	0,184057	0,815943	1,39924	5,71738	1,38451	5,77823
1598	342	1	0,236914	0,763086	1,40123	5,70926	1,38415	5,77971
1599	347	1	0,347022	0,652978	1,3797	5,79837	1,36508	5,86048
1600	301	1	0,453173	0,546827	1,42435	5,61661	1,41801	5,64171
1601	285	1	0,2	0,8	1,43865	5,56077	1,42014	5,63326
1602	296	1	0,0760916	0,923908	1,42007	5,63354	1,38383	5,78107
1603	288	1	0,310803	0,689197	1,40492	5,69428	1,39271	5,74419
1604	287	1	0,254184	0,745816	1,3808	5,79373	1,35911	5,88621
1605	269	1	0,394628	0,605372	1,40126	5,70913	1,37343	5,82485
1606	277	1	0,530655	0,469345	1,4358	5,5718	1,42233	5,62459
1607	280	1	0,320444	0,679556	1,39678	5,72745	1,37468	5,81953
1608	275	1	0,28702	0,71298	1,37903	5,80119	1,35211	5,91669
1609	235	1	0,0289062	0,971094	1,40192	5,70647	1,34989	5,92643
1610	233	1	0,0885942	0,911406	1,4207	5,63101	1,38431	5,77906
1611	229	1	0,309306	0,690694	1,42362	5,6195	1,39096	5,75143
1612	226	1	0,383861	0,616139	1,38935	5,75809	1,3703	5,83814
1613	234	1	0,375506	0,624494	1,39144	5,74945	1,35933	5,88525
1614	201	1	0,322977	0,677023	1,48844	5,37476	1,44324	5,54307
1615	182	1	0,365681	0,634319	1,44178	5,54868	1,41816	5,64113
1616	223	1	0,140582	0,859418	1,44523	5,53546	1,38391	5,78074

```

double res = (w+n) / 2 + (ne-nw) / 4;
int ddvdh = dv-dh;
int ddvdhnorm;
if (ddvdh > 0)
    ddvdhnorm = dv-dh;
else
    ddvdhnorm = 1536+dh-dv;
if (ddvdh > 80)
    res = w;
else if (ddvdh < -80)
    res = n;
else if (ddvdh > 0)
    res = (w0[ddvdhnorm]*res+w1[ddvdh
norm]*w) / (w0[ddvdhnorm]+w1[ddvdhnorm]);
else if (ddvdh <= 0)
    res = (w0[ddvdhnorm]*res+w1[ddvdh
norm]*n) / (w0[ddvdhnorm]+w1[ddvdhnorm]);

```

Для реализации поиска оптимальных параметров разъем исходные изображения на контексты *gapContexts* — класса *GAPContext* в классе *GAPContextArray*. Каждый контекст *GAPContext* содержит множество окрестностей *gapVicinities* — класса *GAPVicinity* из *nGAPVicinities* элементов. Каждая окрестность класса *GAPVicinity* содержит элементы  $x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_9$ , которые буду называть окрестными точками. Таким образом, при подборе оптимальных значений весов  $w_0, w_1$  для каждого контекста не придётся сжимать все изображение, а достаточно выполнить предсказание для некоторого числа окрестностей для каждого *GAP*-контекста. *GAP*-контексты с положительной разницей  $d_v$  и  $d_h$  отображены на номера с 1 до 1535, *GAP*-контексты с отрицательной и нулевой разницей  $d_v$  и  $d_h$  — на номера с 1536 до 3071.

В результате запуска классического генетического алгоритма с параметрами: размер популяции, равный 1000, число генераций до останова, равное 3000, время вычислений составило около 11 часов 19 минут на одном ядре процессора Intel Core i5 6500, были получены следующие выходные данные, сведенные в табл. 1, 2: *nGAPVicinities* — число окрестностей у заданного *GAPContext*; *GAPCR* — степень сжатия, достигнутая оригинальным предиктором *GAP*; *GAPEntropy* — энтропия ошибки предсказания, достигнутая оригинальным предиктором *GAP*; *SGAPCR* — степень сжатия, достигнутая синтезированным предиктором *GAP*; *SGAPEntropy* — энтропия ошибки предсказания, достигнутая синтезированным предиктором *GAP*. Окрестности для *GAP*-контекстов собирались из изображений размером 256 точек  $\times$  256 точек  $\times$  8 бит на точку: *bird*, *bridge*, *camera*, *goldhill1*, *lena1*.

## Л и т е р а т у р а

1. Meyer B., Tischer P. Glicbawls — Grey Level Image Compression by Adaptive Weighted Least Squares // Proc. Data Compression Conf. Snowbird, 2001. Pp. 503—515.

Как следует из табл. 1, 2, именно использованный в оригинальном *GAP*-предикторе низкочастотный предиктор почти всегда (за исключением контекста 1537, при использовании *GAPFLMODE* = 2 достигнут битрейт 3,77932 *bpp*, при использовании *GAPFLMODE* = 1 — битрейт 3,78830 *bpp*; разница — 0,00898 *bpp*) обеспечивает наилучший результат сжатия среди рассмотренных кандидатов на низкочастотный предиктор. Из данных табл. 1, 2 заметно, что дискретный характер определения контекстов *GAP* в оригинальном *GAP* ведет к завышению энтропии ошибки предсказания, так, для контекста 32 разница энтропии, достигнутой в результате оптимизации параметров  $w_0, w_1$  и энтропии, достигнутой оригинальным *GAP*-предиктором, составляет 5,68570 *bpp* — 5,57090 *bpp* = 0,11480 *bpp*, для контекста 33 разница равна 5,70442 — 5,67874 = 0,02568 *bpp*, для контекста 8 она составляет 5,04781 — 4,97770 = 0,07011 *bpp*, для контекста 9 разница составляет 5,14444 — 5,13519 = 0,00925 *bpp*. Отдельно отмечу несимметричность весовых коэффициентов  $w_0, w_1$  для номеров контекстов 1 — 1535 и 1536 — 3071.

## Выводы

Предложена методика поискового синтеза предиктора наподобие предиктора *GAP*. Основную полезность представляет именно методика, возможный выигрыш от использования синтезированных модельных параметров проиллюстрирован в таблицах, но сами значения в таблицах, полученные в ходе реализации методики автором, представляют мало пользы, возможно, из-за сильно ограниченного множества изображений, использованных для поискового синтеза предиктора. Из таблиц видно, что наиболее часто встречаются контексты с малой разницей вертикального и горизонтального градиентов, возможно, наиболее важно определить модельные параметры хотя бы для контекстов *GAP* с малой разницей вертикального и горизонтального градиентов. Ключевыми аспектами описанной методики и важными факторами успеха являлись, во-первых, подбор генератора псевдослучайных чисел с достаточно большим периодом, во-вторых, выделение из тестовых изображений окрестных точек для каждого контекста предиктора *GAP*. Классический генетический алгоритм в результате 11 часов 19 минут процесса оптимизации при достаточно эффективных генераторах псевдослучайных чисел оказался в состоянии подобрать веса модели предиктора не хуже, чем веса модели предиктора в оригинальном предикторе *GAP*, что подтверждает эффективность описанной методики поискового синтеза предиктора.

## References

1. Meyer B., Tischer P. Glicbawls — Grey Level Image Compression by Adaptive Weighted Least Squares. Proc. Data Compression Conf. Snowbird, 2001:503—515.

2. Ulacha G., Statinski R. On Context-based Predictive Techniques for Lossless Image Compression // Proc. XII Intern. Wokshop Systems, Signal and Image Proc. Poznan, 2005. Pp. 990—993.
3. Балашов В.Н., Гольцов А.Г. Моделирование генераторов случайных чисел. М.: Изд-во МЭИ, 2009.
4. Гмурман В.Е. Теория вероятностей и математическая статистика. М.: Высшая школа, 2003.
5. Амосов А.А., Дубинский Ю.А., Копченова Н.В. Вычислительные методы для инженеров. М.: Высшая школа, 1994.
6. Программа для ЭВМ № 2021618408. Поисковый синтез предиктора v1.0 / П.А. Чернов.
7. Карпенко, А.П. Современные алгоритмы поисковой оптимизации. Алгоритмы вдохновленные природой. М.: Изд-во МГТУ им. Н. Э. Баумана, 2021.
8. Чернов П.А. Древовидная модель решения многомерных мультимодальных оптимизационных задач гибридными методами поиска // Вестник МЭИ. 2016. № 1. С. 18—25.
2. Ulacha G., Statinski R. On Context-based Predictive Techniques for Lossless Image Compression. Proc. XII Intern. Wokshop Systems, Signal and Image Proc. Poznan, 2005:990—993.
3. Balashov V.N., Gol'tsov A.G. Modelirovaniye Generatorov Sluchaynykh Chisel. M.: Izd-vo MEI, 2009. (in Russian).
4. Gmurmman V.E. Teoriya Veroyatnostey i Matematicheskaya Statistika. M.: Vysshaya Shkola, 2003. (in Russian).
5. Amosov A.A., Dubinskiy Yu.A., Kopchenova N.V. Vychislitel'nye Metody dlya Inzhenerov. M.: Vysshaya Shkola, 1994. (in Russian).
6. Programma dlya EVM № 2021618408. Poiskovyy Sintez Prediktora v1.0. P.A. Chernov. (in Russian).
7. Karpenko, A.P. Sovremennye Algoritmy Poiskovoy Optimizatsii. Algoritmy Vdokhnovlennye Prirodoy. M.: Izd-vo MGTU im. N. E. Baumana, 2021. (in Russian).
8. Chernov P.A. Drevovidnaya Model' Resheniya Mnogomernykh Mul'timodal'nykh Optimizatsionnykh Zadach Gibrnidnymi Metodami Poiska. Vestnik MEI. 2016;1:18—25. (in Russian).

#### Сведения об авторе:

**Чернов Павел Александрович** — магистр по направлению «Информатика и вычислительная техника», e-mail: pavel\_chernov\_articles@mail.ru

#### Information about author:

Chernov Pavel A. — Master's Degree in Computer Science and Computer Engineering, e-mail: pavel\_chernov\_articles@mail.ru

**Статья поступила в редакцию:** 14.04.2022

**The article received to the editor:** 14.04.2022

**Статья принята к публикации:** 24.10.2022

**The article has been accepted for publication:** 24.10.2022