4.17.95: TM ~~OLHL~~ on [Juergen's [redacted] ppm] on [Learning how to Learn] "OLHTL" ≃ OLH — ELHL — On Long How

I gather that he every once in a while tries a new top level strategy. His evaln. of it is empirical — over t. period in which it was used. How he decides whether it was better or worse than t. last strategy tried, is unclear. — This would seem to be diff/t: hard to define.

[101 + 99 thru 100 Don't exist]

He is certainly trying to get a non-el. view of th. problem & this is to be Commended!

Certainly my own Goal: which is to try to optimize expected future yield, sounds less el., but it has some logical diffys! — e.g. What should be C.B. for estimate of future problems & future ~~year~~ yield on them?

→ ≈ Version Jan 95 p 11 item 14 "If t. task changes"...

p 4 of version N94: He's talking abt "task changes": This can be normally dealt w. by a machine that does not assume a "stationary" seq. of problems, but trys to ~~see~~ regys in t. sequence of problems that allow it to predict what kinds of probs will occur in t. future.

~~Who the~~ "A" of Rivest Shamir Adelman Patent? (RSA)

Impt —

8.1.96: Re: "task changes": I also was considering giving TM a large bunch of problems of diffrnt diffys — most of them well beyond his present capability. It presents of t. more diff/t pbs. gives T.M. an idea of his "direction" of development — ~~wha~~ what he's aiming for in t. long run.

Go to Schm latest website: T. old one isn't so good. — worked as of Nfeb97 — CV: 5 pgs. ∄ file.

look up [Schmidhuber] in AltaVista. [w³.idsia.ch/~[redacted]juergen]

3 597 (Jan 31, 95 paper) — Init (a1, a2) spreads small integers in this region. — ("small" integers (< , series))

-1000 " Min ... Inputstart ≤ inputend ● Registerstart Progstart ... Max " +100

[−9 ... 0 ... 9, <0]

inputarea: from "envt." — registerarea — Prog. Area

Work Area. ≡ work cells.

P_{ij} — inst no. / address no.

I think regstart is at Φ, But register region is $\phi \pm \left(\frac{nops-1}{2}\right)$ & region "nops" in length

Schm. — MMP = .001 — stack size 10 K — Maxint = 10 K

So whole storage is either work area or prog. area.

nops = no. insts. nops << Maxint.

[redacted] T. avg. t. of t. insts are always from 0 to nops−1

Init (a1, a2) (Pages 7 table 1) puts integers up to "nops-1" in work area ~ $\pm\left(\frac{nops-1}{2}\right)$

½ in Reg. area, ½ in non-reg. area. This is in t. context of th. params used in § 5.1 (p 13)

Init (a1, a2) spreads small integers ~~from~~ to addresses about −9 to +9 (i.e. in t. "Register area")

-31

3797 TM OLHL
on Learning How to Learn
(to Lrn)

Learning How to Learn.

10 2
2

P7 of the paper: "a self mod pgm to be "useful", has to start being useful immediately after it ends" — and it can't be unuseful for even a moment! It would seem that this is far too strict a criterion. Doesn't deal well w. noisey systems (which is mankind we have).

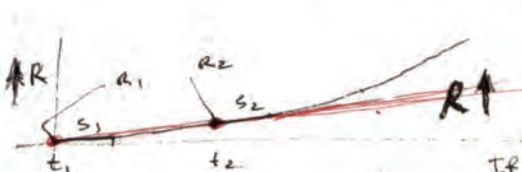
pp7-8 I guess a self modifn pgm consists, ... of the list of modifns ......
But from the dern., the stack would seem to not contain enuf info to reproduce a particular "Self-modn" pgm. Hvr, 8.10-.14 gives one idea that all info about the Pd's — i.e. all changes in the P.d. could be reproduced from infos in the stack. (8.23 may clear this up — i.e. how previous "state" can be restored.)

It would seem that the decision done by 8.22 would be a statistical decision — I guess the "dskP" vector, a list of indices into ... what addresses its from — this is contained in "address" so ... of info so 8.23 can be done.
Tho, apparently not! The R values are taken off the stack; we don't actually rerun the old pgms w. new data!

In 8.22, I'm not clear on which "t" is used: whether it's t for start or end of Self-mod Pgm. I think "start" since the pointer is to $s(s(sp)\cdot first)\cdot R$

Def — The algm. for getting better Samps (Self modifn pgms). seems Hyper "Greedy" to the extent of unviability!

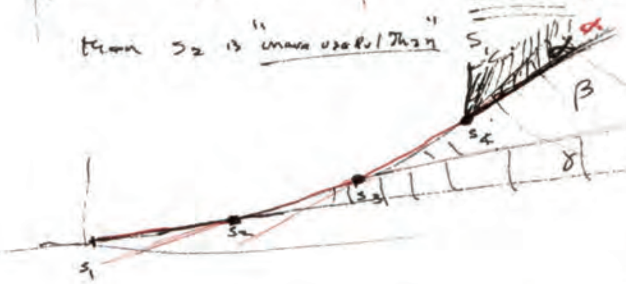
20% dd →
20%/mo.
• xy/yr
15% dd →
15%/mo



s_1 started at t_1
s_2 " " t_2.

20

If, at $t > t_2$, R is above the line $(R_1,t_1)(R_2,t_2)$ then s_2 is "more useful than" s_1



— α is all region above line (s_3, s_4)
If we stay in region α we retain s_4
" " " " β (but out of α) we retain s_3 but drop s_4
" we drop s_4 but stay in γ, we keep s_2.

.24

we draw st. lines thru (s_1,s_2); (s_2,s_3), (s_3,s_4); These st. lines define the α, β, γ regions.
On "Greed": as soon as it wanders from α to β, we drop s_4; we do not consider that "s_4" mite wander back into α again.

Semps = Selfimprovements.

(8.10) A big Q is the "Credit assignment" to various Samps: The apparent rule seems rather bad! We have s_1 s_2 s_3 s_4: If R hasn't ↑ enuf, we drop s_4, but we keep the ΔR that s_4 obtained! Well, maybe not so bad. If s_3 wasn't so good, it will
30 become clear, because R will not ↑ enuf (i.e. at a rate defined by the line
33 (s_2,s_3)).

.35 To what extent does the "unmodifyable top level credit assignment strategy" make it impossl. or difft. for the machine to model the relationship betw R & the Semps? The machine can look at the P.d.'s of itself, but It doesn't have direct access to R & t data. All it knows is: The present P vectors, and that the present system may be better than previous systems, ... but this is not exactly true
→ It does have access to the Most recent "R" value in a special work cell. It has special instr "Read/write" to Read what is in the [V] array; ... unmodifiable.

because of ... 5?

J. may have a point that I didn't quite get. But I'm not sure that he has. The overall goal of the machine is max mean ↑ in G per unit time for "rest of life". I felt that the finite horizon for the future was hazy — or exponential ↓ in w.p. as one → future. (like future value of money).

— That w.o. a definite T, the goal was not well defined.

One could have considerably different machine behavior, depending on how large ↑ the future was. (This was my "future oriented machine" (FOM).

W. small ↑ the machine was not so interesting. W. large T it was, because it could afford self improvement not immediately oriented toward immediate gain.

∑ like a corp. w. research oriented to (1 yr horizon U.S. 20 yr. horizon Japan).

I think I was concerned with the idea that if T were very large, the machine would rarely get around to working on problems that had any immediate payoff — or any payoff at all!

The division into working on present problem U.S. working on "self-improvement" seems to be very artificial: which is a simple pt. that J. brings out. In a final TM, there would be no such "division of work". However, in earlier designs, it may be useful to adopt such a "crude" approach.

4.3.97 [redacted] (~2 years after I rcvd. J's L.H.T. Learn paper!) Another point!

J. is concerned that TM be working on the "latest problem" at all times. Maybe this is not so good way to manage: At "home learning" I would spend much time on interesting puzzles I discovered, & cut-off new projects until I understood them. I colleagues I resent didn't have this luxury, so I stored many "intriguing ideas" for "later" — & "later" may or may not have arrived: The idea is that a TM has a choice of making repeated trials on a particular problem before deciding (or deciding not to) move on to new problems. Depending on the learning environment, TM may or may not be able to "afford" this alternative.

In general, a machine will have/a certain rate at which innovation in problems can be introduced. Introduction of innovation at a rate >> this will give the machine little chance to learn. (like giving a newborn baby a calculus problem). On the other hand, too slow an innovation rate is wasteful of the machine's time.

One could introduce new problems before the machine completely solves older probs — if one has a good way of giving "partial credit" for partial solns. Another way

to deal w. TMs: Give credit only for complete solns. Don't ~~give~~ ask TM to solve new probs until it has solved old ones. We may make exceptions if TM is taking too long on a problem -- suggesting that "prob_i is too hard for TM's present state of development.

4.4.97: On more learning (TM_2): I was having some dirty w.r. practical problem of improving TM_1's pgm: It's not necy to do many experiments to self improvement: It's like trying to ride a Bicycle: One gets some empirical info. by trial & error: Then one "thinks about it for a long time", then does some more experiment, then thinks a lot ... Most of time is spent "thinking". Experiments are expensive in time & other resources.

In general, w. finite TL & finite future, the main goal of TM is to improve TM_2: — i.e. to get the max ~~amt~~ amt. of property amortized G for the future. This would involve deciding how much time to spend on the current problem: (which contributes Much to future G, because it gets lots of wt. (since Δt is small).

His 2.31, 33, 35 has 3 ideas he feels are very novel in his system.
1) Adaptive modifn. of Levin Alg. 2) Learning is from next-nearly repeated events.
3) Objective top level gorc. considers cc of learning (i learning how to learn).

(SN) Some Remarks on Levin:

.22 1) What about random choices w. probys $\propto p_i$ (or $\sqrt{p_i}$?) w.o. replacement? Some time will be wasted verifying that already "tried" cands have been tried. Every once in a while one doubles (or otherwise increases) the proportionality factor: cc_i limit is pc_i.

random Levin: random selection of trials, but w.o. replacement

Should sometimes be random, w. probby $\propto p_i$ or $\sqrt{p_i}$? How bad it = probys used? (x exhaustive rule such).

.25 2) In .22, How impt. is selection probby $\propto p_i$ or $\sqrt{p_i}$? would = probys over all cands w. probby $> \epsilon$ a certain ϵ (which ± 2 each round) be o.k.? If so, then rote trials (non probilistic) would be equally good & have ad vantage that "non-replacement" is no longer a problem.

Well: consider .25: Say we have gotten to the round at which cc_j is enuf for cand$_j$, which has $pc = pc_j$. $\frac{cc_j}{pc_j} = T_j$; other cands will have $cc_i \leq T_j \cdot pc_i \leq cc_i \leq T_j \leq pc_i$ (1 $\leq pc_i \leq 1$) so $\leq cc_i \leq T_j$

fine! No apparent advantage to doing trials in pc_i order.

We do have find way to pick trials w. $pc_i >$ some threshold, ϵ

→ See 7.2 for how to do this & also how to do pc_j ordering & cc_j ordering

.34 I think this is not hard in many cases, but I'm not sure. — In case of computer pgms, it merely means pgms of $\leq$ length $< R_k$ — but for other probilistic situations, it's not so easy!

Say one has m objects in the j one can have n_j distinct probby values. (not all necessarily distinct)

I see a kind of recursive soln:

.01 Search, contn Let a_j^k ($k = 1 \mid n_j$) be the probes assoc w/ the k^{th} object of type j.

Let $b_j^k = \ln a_j^k$; Then ~~we want to take 1 each of 1~~ for each j we want to take also $b_j^{k_j} \Rightarrow \sum b_j^{k_j} < E$. ~~How~~ What are all ways to do this?

Well, say we had an algm to do this! Then the algm is recursively definable in following way:

Pick ~~one~~ smallest of ~~all the~~ b_1^k ($k = 1 \mid n_1$) ($\equiv b_1^{k_1}$). ~~Then~~ Re-index in order of size of b_j^k.

.06 Then solve the problem for $m-1$ and ~~even~~ $E \to E - b_1'$

Well, this problem is best solved backwards. Perhaps it could be worked w/ a stack.

What is wanted is "boundary configs": Select a $b_j^{k_j}$ for each j such that if one picked any of the $b_j^{k_j}$ larger, $\sum_j b_j^{k_j}$ would be $> E$.

For each boundary config, one has a bunch of simply connected cartesian products, that sum to $< E$.

So: to find Boundary configs! The boundary conditions are also defined recursively (like .01 - .06.

b.c. Another approach! Given any (b.c. / bndry config) for $E = E_0$, we can perhaps find other sets of other b.c.'s with E minimally $> E_0$.

.20 ~~enter~~ AH! a reasonably practical soln: Say $b_j^{k_j}$ is the smallest of the b_j^k ($k = 1 \mid n_j$) define a new set of objects $c_j^{k_j}$: ~~[illegible]~~ $k_j = 1 \mid n_j$

$c_j^{k_j} \equiv b_j^{k_j} - b_j^0$: ~~[illegible]~~ c_j^1 is undefined.

$c_j^1 \equiv \phi$. always

It is easier to make ~~[illegible]~~ b.c.'s of $c_j^{k_j}$'s, because we can always assign $c_j^1 = \phi$ to some j. $E \to E - \sum_j b_j'$.

Recursive Soln: So start with $j = 1$; ~~[illegible]~~ $c_1^{k_1}$ can be any value $< E$.

for each value $c_1^{k_1}$ picked; do $E \to E - c_1^{k_1}$; Then solve the assoc $(m-1)$ place problem for $j = 2 \mid m$.

This is a nice soln, because for each choice of a $c_\ell^{k_\ell}$ value we may $\downarrow E$ (or not if $k_\ell = 1$) & we know exactly how many possys there are for $c_{\ell+1}^{k_{\ell+1}}$.

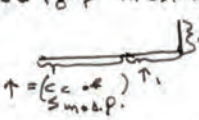
↳ I think this ↑ recursive soln. can be realized w. a single stack in an efficient way. 7.27 ← v.g. see 7.27 for very easy & efficient, practical w

.30 .31 What is the "Markovian interface" p10 item 2; 15.35; 2.35 18.27 8.10 "H Q-Learning ... Neo Markov... " first linear paper; But I don't think the explanation

.33 How is "credit assignment problem" - 18.30 check relevant? 17.37 9.25 {He may learn different things by Cmod. Ass. prob.' Now what I mention 'normal usage'". 6.02

Im: In what sense is his system oriented toward the future? Its criterion for goodness of a simp is its effectiveness recent past.

Smod pgms must be rather short! since If they are long, longer pgms must have very large ΔR per unit time!



Then Q of this smodp is $Q = \frac{\Delta R}{\tau + \tau_1}$: if τ is large, Q will not be

Very large. Early in this TM's life, it will arrange its params so that Smodp's tend to be

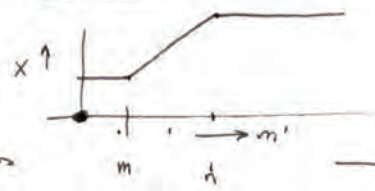
01 Short. As TM matures, it may slowly allow longer & longer $\langle$mod p's.

02: 5:33 Credit assignment (strategy prob): I think what he means is "After a OR occurs! How do we respond to this new info? How do we modify the system? What "parts" do we change & how do we choose them?

See A General Method ... multi-agent learning" & 6.22-.32 for his discn. of "learning from example only" Also J's & Wiering's L_{SRCH} paper is on "Markov decision process"

07: 5:30 : on L_{SRCH}: On what happens when the p.d. changes during L_{SRCH}; or between "$T \leftarrow 2T$" runs?
If p.d. changes betw. $T \leftarrow 2T$ runs, we will get a different soln., if the change occurs early enough! The drop-out criterion is $\frac{T_j}{p_{ej}} > T$ (where $T = T_0 \cdot 2^n$ for nth pass run)
Suppose that w.o. modifn. of p.d., soln. would occur on nth round.
With modified p.d., it occurs at mth round ($m \leq n$).
If a modification occurs on the m'th round, then time for each i is $T_0 \cdot 2^x$:
if $m' \leq m$; $x = n$: If $m' \geq n$, then $x = m$: The p.d. modifn came too late if we solved the prob w.o. it.

Possibly use a combination Disc Cache for both time — & hand L_{SRCH}?

.17 

If the discovery of a p.c. of soln. occurs during a $T \leftarrow 2T$ run, whether it helps or not depends on whether the solution correct soln. was worked on before or after the relevant p.d. modifn.

.20 → The more exact way we would modify the p.d. during a run, is unclear

If we do time sharing on both the direct L_{SRCH} (Inv. problem) and the "p.d. improvement problem", then, first, we run into remarks (.20). Then, part of the soln. to the problem will have been done at higher cost than it should! Ideally, we would want to re-allocate probs in the past, as soon as the p.d. modifns have been decrd.

Hmm, the ability of modifying p.d. during the L_{SRCH} would seem to be considerable (but not nearly so). In the time-shared mode, this is the only way one can modify probs! — Anyway, for the time being, probly it's best not to use Time Shared L_{SRCH} (Bc it is at least 2 times as fast as $T \leftarrow 2T$ L_{SRCH} (& probly significantly faster — as a function no. of levels of complexity of the soln)) — Since $T \leftarrow 2T$ is simpler & will illustrate the general principals.

Anyway, the Big Q was: To what extent is L_{SRCH} for OZ problems optimum?
In the case of optimality of L_{SRCH} for INV problems, if we can express all hours as modifns of the p.d., then, presumably, w. the right p.d. (i.e. including all needed heuristics), one has the optimum soln.
In the case of OZ probs, can one say the same thing?
Can we apply the (second/first) G-House Thm. to OZ probs?

The optimality is based on the "Gambling House Thm": That for fixed p.d. $\frac{c_{ev}}{p_{cz}}$ ordering is optimum.

Actually, in OZ probs one has an a priori in optim. methods. For normal OZ probs this a priori is fixed. I think in general, one gets experience w. optim., so after looking at a prob. — studying it a while, one obtains an a priori over all optim. methods that is peculiar for that problem. It is conceivable that this a priori would change during the soln. of a OZ problem — just as in Inv problems. It is conceivable that this a priori would change during the soln. of a OZ prob to the p.d. over all optim methods — but I think this is unusual, because the relationships of an OZ prob to the p.d. over all optim methods would change infrequently (slowly).

→ 7.21 sheet.

4.5.97 : TM : LHL

In writing to J about his rounds-scheme ("credit assignment"): maximum ≤M problems: in which 107 →
ln(mult. yield at ln(1−ddn)· for 5% ddn say ×1.95/yr. 10% ddn → ×1.3225/yr.
Hvr. 0% ddn → ×1/yr. (roughly) 15% " → 1.52/yr.

It is possl. to have >×1./yr with zero ddn, but only by not using inst. ddn. —
Which is a bit deceptive.
Note that this SM type problems are typical of a broad class of problems in R.W.
It may be that for the "noiseless" probs he was using that "zero ddn" won't work.

One way to do it: a TM allows a smod prm to stay if its rate of return is a certain % (say 25%) below previous "best"; but this is allowed only for a certain time interval, T. For anything past T, we expect it to be better than last "best" s.mod prm.
The "25%" & the "T" can be decided by the smod prms. — Tho we may want to craft
out TM w. some apriori chosen values to speed up convergence.

Actually, it's probly possl. to do a good stochastic analysis to find the best strategy to use for discarding "smod prods."

Probably a better way would be to look at all previous smod prms & how well they did (not just ones that were kept).

.21 : 6.40 : Hvr., computing the probs of various optzn methods as a part of the OZ problem could take a lot of time! — So perhaps, during one or more of the OZ procedures, info would "come out" that would cause one to change the p.d. over optzn. methods.

.22 just as in OZ probs! In INV probs, the p.d. should also be a funct of the problem — ⇒ 14.01
Unless one could somehow automatically include this in the srch. method! — in which case one
.26 may also be able to do it in OZ probs!

.27: Srch Desc IN General: In probabilistic prms. like J's, it seems very difft to put prms in pc or pc/cc order. One possl. way: (see 5.01-30 for some relevant ideas) 5.31 Marginal 6.08

Anyway, it's like prmg. a Turing Machine w. a taper w. slight modifn: We do sequential prms, & we note
−ln p of each instruction, and we plot the most likely at each step, until we exceed p threshold or
pc/cc threshold. We do this search via a stack, and we do a complete "breath first" search —
(By "breath first" I mean do all branches at each inst one encounters, until Least limit is reached)

This can be done via simple to order probs of simpler kinds as well, like 4.36ff
GOOD! This is essentially the "Lsrch" that I wrote about in my "Opt. Sap. Srch" report.

Note that the prmg. listing includes partial prms (i.e. incomplete prms): This can be wasteful; if the time to do each step >> the accuracy times, one may want to list all complete prms. This might be done using the technique of 5.20, of dividing out the probty of the most likely choice (≡ largest pc).

→ See 22.01-33 for summary of Proof

$E \to E - c_i^{R_i}$
$e^E \to e^E / e^{c_i^{R_i}}$

4-5-97 TM: LHL

.01 I've been thinking about J's "ultimate Gore" incorrectly. If a smodp occurs & it continues to be used, it will have a certain Q = mean ΔQ per unit time. If no new smodp's are used it will continue w. same $\frac{\Delta R}{\Delta t}$. If a new smodp [redacted] doesn't ↑ that $\frac{\Delta R}{\Delta t}$, then its not doing any good.
If t. cost of creating a new smodp is zero, then we should discard t. old one, since its expected Q is < 0 & t. expected Q of a random smodp is zero (∴ better). Hvr., creating a new smodp does have a ≈ known certain time (from historical data).

Suppose that Q for new smodp's has an a priori d. about zero: — a rather narrow d.f.: small σ, $= \sigma_0$: ~~This is rather indic of how many trials~~ Also, due to problemt nature of pgms, when we get a series of ΔR values following a smodp, the value of t. long term Q is known to be
$\sim \bar{Q} \pm \frac{\sigma_1}{K}$.

So; After one R after a smodp, we have a certain Q: it is appears to be ϵ below previous

.0cp Q (which is also uncertain). [curve] $\to \Delta Q$ t. Expected ΔQ obtained by t. smodp is $< \phi$, — But t. d.f. is rather broad. The probability of its having a true $\overline{\Delta Q} > 0$ is $\int_0^\infty$ of t. displaced Gaussian. It is slightly $< \frac{1}{2}$. $= \frac{1}{2} - \epsilon$ On t. other hand, if we try a new smodp, we have a priori a proby of $\frac{1}{2}$ of its being eventually good. ($\widetilde{\Delta Q} > 0$).
But this new smodp has a certain time cost Δ_1 overhead $= \Delta_0$ & t. present smodp has no overhead.
~~So the cost of new experiment~~
So, say t. next R occurs at time Δ_1.
So expected value / of present experiment is (divided by cost) $(\frac{1}{2} - \epsilon)/\Delta_1$
" " " " " now " " $(\frac{1}{2})/(\Delta_1 + \Delta_0)$

.22 So if $\frac{\frac{1}{2} - \epsilon}{\Delta_1} > \frac{\frac{1}{2}}{\Delta_1 + \Delta_0}$ Then we don't do / now smodp yet.

$(\frac{1}{2} - \epsilon)(\Delta_1 + \Delta_0) > \Delta_1 \cdot \frac{1}{2}$ ~~$\frac{1}{2}\Delta_1$~~ $+ \frac{1}{2}\Delta_0 - \epsilon\Delta_1 - \epsilon\Delta_0 >$ ~~$\Delta_1 \cdot \frac{1}{2}$~~

$\frac{1}{2}\Delta_0 > \epsilon(\Delta_1 + \Delta_0)$ time overhead for new smodp; Time to new R.

$\frac{1}{2} \frac{\Delta_0}{\Delta_1 + \Delta_0} > \epsilon$

$\frac{1}{1 + \frac{\Delta_1}{\Delta_0}} > 2\epsilon$

$2(1 + \frac{\Delta_1}{\Delta_0}) < \frac{1}{\epsilon}$

$\frac{\Delta_1}{\Delta_0} < \frac{1}{2\epsilon} - 1$

$\Delta_1 \sim 1000$; $\Delta_0 \sim 2$ or at most 10; unless large administration costs. ∴ so $\uparrow \approx \frac{\Delta_0}{\Delta_1} > 2\epsilon$
After we do several trial w. t. current smodp., ϵ will change, or $\frac{5}{1000} > 2\epsilon$
so don't get rid of smp if $\epsilon < .00025$ — True for d
I don't know if t. $\frac{n+1}{n-1} \cdot \sigma^2$ correction is relevant.

estimate Δ_0/Δ_1 into around p 14.

.31 1k time steps per payoff event. 9 k proby modifies in 10^6 time steps; so 9 per 1000 time steps
~ 1 in 100 time steps. After 10^7 time steps about 4 modifs/1k time steps. 1 zero → 6.6 / 1k steps.

.37 (SN) Since t. info in "work area" is not reinitialized at each run start, it could contain any [redacted] possl. memory of t. system. [redacted] — including info about smodp's that [redacted] were no longer "useful" ⇒ 13.05 13.07

4.6.97 TM : LHL

Bad SMP

A Q about How T. [illegible] system works: When one discards a (smp) as not "useful": Self Mod No. Pgm.

Does one also discard the R values that it worked in? : It seems reasonable to do so.

Look at LHL (P7): Q(s,t) is defined, act.

From P7 I get the idea that the R values are not discarded. If a "useful" smp is followed by a very long string of non-useful smp's then the Q of S_0 will ↓. — possibly enough so that S_0 will have to be removed.

After a smp has just been removed, is there usually a long period until a new smp is tried? from my 8.31, ~5 smp's in 1000 time steps.

108 A disturbing fact: as time goes on, the contents of "work areas" changes: So the significance of older smp's changes. → 13.05, 07.

How do smp's start? They supposed to end. whenever the "end smp" inst. occurs; (perhaps they start as soon as one of the 3 self-mod insts. is written! IncP, DecP (Garp "perhaps too.

yes! 7.06 ! 9.34 ; 3.14

Maybe "life's events are non-repeatable"

These ideas are termed in 3.14 of the paper. "Life is one way" meaning

"Induction based on single experience." meaning? 10.04 ; 11.33, 34 ; 8.10, 11 ; 11.50

The method of probability assignment to pgm calls would seem to make Lsrch drift! I.e. hard to order insts in PC or $\frac{cc}{pc}$ order. (NO!) ! use 7.27 ff

The time rest of a smp has to be included in computing its "slope", Q, because during it smp, the machine could be doing things that directly contribute to payoff!

Actually it does: see 1.25 ff!

Unfortunately, the fact that there were only ~5 smp's per 1000 (time steps) doesn't tell one much about the lengths of smp's. Dependence probs of IncP, DecP v.s. probs of end self-mod

.25 We know prob of "end self-mod" is $\sim \frac{5}{1000}$: that only 1 or 2 self IncP, DecP's occur per smp suggests IncP, DecP be each have about same prob as "end self mod", so if there 3 insts occurs at random w. prob .5% each; What is expected time difference betw. (IncP, DecP); (ends mod)? I would guess ~ 2% or 1%:

Poisson distributed. So say P(IncP) = α, P(DecP) = β, P(end self mod) = δ = .5%.

I could run a Monte Carlo to get mean time betw. δ's (i.e. α or β) & betw (α or β) & δ.

If a δ occurs, αβ occurs about $.5\%^2 = .25\%$ = 100 into future. After that it's $\frac{1}{.3\%} \approx 200$ time cycles before δ occurs again and 100 Inc Dec 200 and. So smp's tend to be ~200 time cycles long and separated by 100 time cycles: Because 200 & 100 are poisson distributed.

.30 This gives good values for 8.22 - .31! $\Delta_1 = 1000$; $\Delta_0 = 200$: $\frac{1}{1+\frac{\Delta_1}{\Delta_0}} = \frac{1}{1+5} = \frac{1}{6}$ so $\frac{1}{12} > \epsilon$ is criterion.

Actually it's more complicated; 2 "end"'s can occur in a row.

Actually a 2 state machine w. poisson transition probs.

State (1) Inc or Dec has just occurred; state (2) end has just occurred.

Say P_1 is prob of Inc or Dec P_2 is prob of end.

Mean time of State 1 = $\frac{1}{P_2}$ since

State 1 self-mod: $T_1 = \frac{1}{P_2}$; State 2 betw self mod: $T_2 = \frac{1}{P_1}$

For there to be 5 smp's per 1000; $T_1 + T_2 = 200$ time steps. $= \frac{1}{P_2} + \frac{1}{P_1}$

For there to be ~ 1½ Inc or Decs per smp: $P_1 \approx P_2$; $T_1 \approx T_2$ so $T_1 = T_2 = 100$ time steps. → spec 10.32

4.6.67 ! TM:LHL: **HINTS: (2 OR)**

Another way to think about t. system: It just runs along producing its ΔR's * ——

Also self mods: Then "end self mod" comes along in a Poisson way to segment this data string into (smp's) —— and Dec, Inc

well be! because each time a smp is "useful" its probable changes are restored.

Perhaps write to J about t. importance of sharpening crudeness of a "benign external envt".
Very impt. in biology: Man goes even further in developing clothing & housing to further control of internal envt. Similarly, with problems coming from "outside world": Man selects certain of them to work on, me simplifies them into "models"
Then he makes many deductions on the same "model".

Q about system [redacted] operation: smps take ~ 200 "clocks": Actually is over 1000 clocks.
So normally, there will be no change of R during or for quite a time after the smp has been worked on. So normally, since t ↑ is R is constant, we expect Q to ↓ after a smp, until the new ΔR arrives.

It's rarely many smps!

seems kind of silly not to wait for the new ΔR before deciding whether to retain the (last smp!) → 11.12

.20 Somehow, I haven't been considering certain info available in the work area!

i.e. [Ip, Sp, T (models)] The Sp tells where the models of the probys are.

So we can study them & maybe usefully modify them.

Another call has the value of last / known payoff.

Ex: from early part of §5 (P13): "The external envt" is, (I guess) the

Special set of 30 vars V_i (2:d|35): "External" means (I guess) that

it's not part of the regular "work" area. We could (maybe as well) use

another region of the work area for this, & we wouldn't need the special

instrs write, Read (Table 2 P13).

The idea of a [V_i] being special is Read, write being

Special, may encourage TM to use them in special ways —— since they are,

indeed, "special".

A "HINT" is a problem-specific Heuristic:

Hints reduce cc of soln. by giving probabilistic info that reduces search cost.
They can be domain-specific, or not domain-specific (very gen.)

"Hints" are problem-specific — reducing cc of soln, but, hopefully, getting clue/clues to do some useful search, nevertheless (ecs).

Hints run from None to all the complete soln. of problem or Negative Hints & cc of soln!

.32 : 9.40 Again about t. "All over credit assignment" smp's! Smp's are probabilistic objects: eval/credit usually Monte Carlo evaluate them on the basis of a trial! Sequential testing (Wald) is appropriate — Drop the test after the expected yield (for present smp) is < expected yield for new smp.
This is the idea of 8.01 ff; 8.10 ff

Of course the idea of MC evaluating of the probabilistic object, is of questionable value — i.e. is it an efficient way to test it? There may be a better Lerch-like way to restrict. (see 7.23 ff!!

.01 Is t. V register reset after each pgm is is re reset or is V reg reset only when ΔR occures? (when)

It has to be when R occures. (see J : 13.30) : Yes.

1) Assume J computed E Q's correctly — then is it O.K.? T. problem of RTM; h=1?

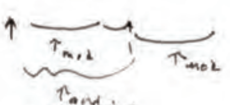
(SN) HM! Say $P_{end\ of\ smp}$ is very large. Then $P_{incerpt}$ is small ≈ $\frac{1}{200}$.

In this case, t. length of a smp tends to be quite small. i.e. "end of smp" inst. occures very frequently. Trouble is, this would give a mean no of proby modifns per smp of close to 1; rather than 1.5.

I may have been wrong in 9.30: Δ_0 should maybe be 100 not 200; but much more serious:

Most smp evaluations occure befor a new ΔR is given, so we really know nothing about the Q of t. latest smp! If the Q increases to acceptible level rite after a ΔR, then just befor that ΔR, it should have had an unacceptable Q level!

.12 In fact, unless a smp straddles at least 1 ΔR, it will have a Q of ϕ, & will be discarded as soon as it is completed. If smp are of average length, 120 — which seems to be about rite — Then



$T_{mod} \cdot 2.5 = 200$: $T_{mod} = 80$; $T_{end} = 1.5 \times T_{mod} = 120$

so mean smp length is ~ 120

≈ smp length only $\frac{120}{1000}$ = .12 or ≈ $\frac{1}{8}$ of all smps are given Q ≠ 0.

so $\frac{7}{8}$ of all smp trials are simply wasted!

Note, here, that the **ΔR's** are not "1 shot evaluations". During t. 1000 "time steps", the pgms being evaluated run many times: (They usually run many times during a smp) ← (Check this!)

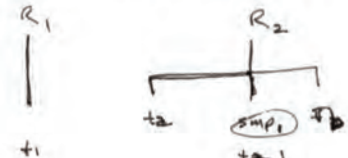
So the ΔR's give an average goodness of t. runs during this 1000 time steps.

Hvr, its not a simple average goodness since t. max ΔR was 30.

Anyway, The ΔR obtaind by a smp usually covers many runs, so i.e. $t_h \cdot s \cdot s z i s >> 1$.

t. ΔR's σ is not nearly very large! The analysis of 8.09 ff is probly much in error quantitatively — but that general method of analysis is still correct.

However J's evaluation of Q is wrong:



The Q of this smp at time t_2 is

$$(R_2 - R_1) \cdot \frac{(t_2 - t_a)}{(t_2 - t_1)} \cdot \frac{1}{t_2 - t_a} = \frac{R_2 - R_1}{t_2 - t_1}$$ which is indep of t_a.

Seems wrong but is actually correct!

Well, If t. smp did not occurr, $\frac{R_2 - R_1}{t_2 - t_1}$ would be expected to be Q_0, t. Q_0 of t. last useful smp. Any difference btwn $\frac{R_2 - R_1}{t_2 - t_1}$ and Q_0 must be atributed to the (new) (smp_1) &/or Noise. If $\frac{t_2 - t_a}{t_2 - t_1}$ is small, smp_1 will have much less effect on $(R_2 - R_1)/(t_1 - t_2)$. The criterion for keeping or poping smp_1 is $\frac{R_2 - R_1}{t_2 - t_1} - Q_0$ it is indep of t_a

It mite be that that since smp_1 is "working" from T_a to T_b, its effect would ↑ as it works: But in fact, the first self mod inst. occures at t_a, & usually there is no second onwards of time is there a second self mod. inst. & this tends to occure late (near t_b).

If smp_1 covers > 1 ΔR, use the large Δt's implied.

4.7.97 TM : LHL — " = More generally, the way want t. system to discover how to deal w. this big workspace trials !

In general, J is evaluating t. smp's poorly ! A much better way would be to do a ΔR at the start of the smp, Then on when t. next smp starts. The ΔR during this time would be a much more meaningful evaln. of t. smp.

J would say " But we have no control over the ΔR's : They are from t. external envt. " (some within this)

Another way : after a proby change, system → "state 1" It remains in state 1 until ΔR occurs ; After ΔR → state φ ; At state φ "end smp" becomes a possl. inst. (otherwise illegal) | T. main trouble is 1 only one smp per ΔR : I don't think it's possl. to get more info than that !

More disturbing comments ! 8.33 : ← these involve "T. workspace" & its uses a non-resetable memory. (see also 9.08, 13.05 ff.

In view of 7.27 – .40 ! Review t. problem of modifying proby values during Lsrch — say during Timeshared Lsrch as extreme case : But in other cases as well. —

Say during a TEST run

Reread pp 1 thru 8 then 13 thru 16, to be sure I understand all aspects of the system.

N.B. The things in t. system that are not remembered

1) Reset of V⃗ to φ⃗ after each ΔR (every 1000 timesteps)

2) Reset of a few (usually 1 or 2) components of t. "P[illegible] proby vector" to an earlier state, if t. last smp should become "unuseful".

The system has access to IP, SP, t mod 10000 : I think it also has access to latest R value, but I haven't been able to find reference.

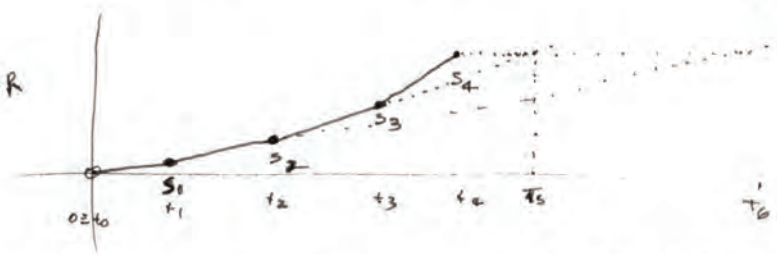
A perhaps easier way to understand J's "Top Gore" !

| | | | |

we look at t. mean ΔR since t. start of a smp. Compare it w. t. mean ΔR since start of its previous smp. If we don't have an improvement, we pop t. stack &

Going back to graphs of 2.20 !

At t_4, even if ΔR = 0 out to T_5 we would not pop. (we would pop s_4 quickly, but s_3) w. ΔR out to T_6 we would pop s_3, but not s_2.



This stack popping is mindful of the search of a tree via Lsrch ! (TM c.p. 7.27) ! we go back to a (branch pt. s_i) at which things were o.k., & then we try new modifns of that branch pt. Hvr, since this is not Lsrch, we do such "w. replacement" ; we don't try to avoid low score branches of t. past : Hvr, t. space may be so large, that a random choice has little likelihood of getting near a previous successful trial.

(SN) Note : T. slope in (.30) is t. true "score" : its t. mean no. of correct V_i per unit time.

6:32:00

206:
646
5104
6:39

User's guide,
(till July 6, 97 from Jim

2.1 Gig should work.

Virus? 6:55

6.22

Full upgrade.

6.
620 :

6.22
Full version needed : 3 discs.

try running all 6.22

Roll V
Mon N1130p
"
Ch 38

4·8·97 TM : LHL :

There is a tendency to think of the system as a hill climb in pgm. vector space : the space being more continuous than the deterministic pgm space ~~has a pod~~ on which it is a p.d.

Hwr., there is more going on: At each point in time, we have the present prob vector, & the present ordered set of "useful" prob vectors (the path in prob space).

.05 The additional thing is not only the prob vector (present state) but the contents of the work space. — which is never reinitialized.

.07 So : a big Q: To what extent is that memory in the work space ("=" "Environment") used? Would some relatively simple hill climbing techniques in the prob vector space be as good or faster?

Perhaps not : Probably the system would soon discover that only instructions near the end of a pgm. are important, so it would narrow its search alot. (The this could also be discovered by many simple hic. methods)

Actually, this may not be true : Suite 501
Apparently the pgm size is varied only from +9 to +100 or 91 insts. long. It could store several possible short pgms which probabilistic jump choices being made near the start of the pgm.

Hwr, due to the (probable?) universality of the inst. set, its pretty hard to guess what the system is likely to do. We have at best, a small no. of ways that we write pgms. This system is probably not even mildly restricted to use anything like those ways! Hwr, Then A Big Q is : is the search space so large that we won't usefully solve any problems? Also, we still have to write training seqs (or equiv').

very exciting if true!

5.. for: Big · ~~Q~~ (1) How "work space" many influences system 8.33, 508 : B.05

(2) The system seems to be "honestly" predicting the future : Is it a RTM w. $h=1$ or $h>1$ or ~~h=1~~ say $h = 2T$ for a system at time T.

(3) Could this system be implemented w. Lsrch ? : Apart from the Gödel dirty of it not really working on Lsrch like problems! It looks like the space is usually to Big to do Lsrch on; That the reason probabilistic search works is that there are very many "not bad" solns in the space.

One could do an "Lsrch" to find a pgm that would get $\Delta R = 3.0$. This is much different in spirit from what J wants to do & the problem he poses. He wants to have an anal. that occasionally recognizes TM — so its an OZ problem! The data is, perhaps, a seq. of pgms that TM tries, plus the total ΔR for those pgms : Data on ΔR being given "occasionally".

The OZ probs are solvable by Lsrch, I don't know of any that have been solved using the "optimum soln": It is, hwr, possible to solve them using "INV. hillclimbing".

.01: [7.21–.26] On Lsrch for OZ problems: In Inv. probs: If someone proposes a soln. better than t. Lsrch w. a particular a priori pd (d.f.), I say "We ~~should put~~ have not put 'all' info into t pd. — That t. new apparently superior method, had info not in t. p.d. — That if that info was put into t. p.d., Lsrch would be as good as this "apparently superior method".

.05 Can we use this kind of argt to show superiority of OZ Lsrch? As in the Inv. Case, it would also tell us just what kind of info we need in t. p.d.! [just what argt. t. p.d. has.]

.07 Another Argt. from Inv. Lsrch: Any heuristic idea, any method of solving an Inv. problem, must if legitimate, come from induction of solns. of previous problems — This assures us

.09 That it (should be / can be) put in probilistic form.

I think .01–.09 may be all we need to prove true optimality of OZ Lsrch!

(Well: There are t. Gamb. Hs. Thrms. for Inv. problems: We have upper bnd in cc of soln. for Lsrch: for Inv. probs. We have actual min Expected cc of soln. for truly optimum srch. Do we have any thing like these G.H. Thrms but for OZ probs?)

Well, O.K.: Say we had a p.d. on all OZ prob. solvers. Given an OZ problem: $M(\cdot)$ [it gives Gore of its (string) argument.]

$P(M(\cdot), S)$ is a p.d. over $(M(\cdot), S.)$ S is t. space of all OZ prob. solvers.

What does this $P(M, S)$ mean? Presumably it was obtained so as to Maximize (expected Gore per unit time) I should also add T to put it in classical form.

[A later development — "Anytime problem."]

So in old form $P(M, S, T)$ is a p.d. ∋

Given an a priori pd on M ($P_1(M)$), Then if we apply S_i to M for T we have a certain Gore at t. end of time T = $G(M, S_i, T)$

Given $P_1(M)$; $P(M, S, T)$ is ∋ $\sum_{P} P_1(M) \cdot P(M, S_i, T) \cdot G(M, S_i, T)$ is maximum.

This defines $P(M, S, T)$.

I'm not so happy about dependence on $P_1(M)$! try again!!

.29 $P(M, S, T)$ is ∋ $\sum_{P} P(M, S_i, T) \cdot G(M, S_i, T)$ is max. [summing over all S_i in S.]

Ordinarily, we don't know $P(M, S, T)$, but we infer it approximately from previous experience w. OZ problems.

If T is not given, it's not clear how to define things.

Say we are given M but no T. We pick a T value, and work on increasing $P(M, S_i, T)$. As T approaches we double T, and continue. Or, use a T value That is always k × present T. (k = 2 or 1.5, say).

In t. defn. (.29) of $P(M S T)$; is There not an assumption of linearity in t. Utility function, G? (linearity wrt what?)

Given that .29 is true of $P(M, S_i, T)$: what is t. best strategy for solving (M, T)?

It would seem best to pick a single S_i ∋ $P(\widehat{M}, T, S_i)$ [find] is max, & run it for Time, T.

4-8-97 TM: LHL:

In contrast using time shared version of Levin's, we spend a fraction $P(M,T,S_i)$ of our time on optim. method S_i. [redacted] If $\tilde{S}$ is the cheapest [redacted] soln. method, then after time $\frac{T}{P(M,T,\tilde{S})}$ we have spent a total of T on $\tilde{S}$ so we have its "optimum soln" at a factor of $\frac{1}{P(M,T,\tilde{S})}$ more in cc.

It would seem that if one had a "better" strategy for (M,T), it would be because one somehow was able to assign a higher prob to $\tilde{S}$ or assign a very high prob to a S that was not as good as $\tilde{S}$ but whose $\frac{S(M,T)}{P(M,T,S)}$ was smaller than that of $\tilde{S}$.

Maybe in analog w. GH Thm: for each Yield, [redacted] $= Y_i$, one has p_i of winning. So [redacted] cmpte trials in $p_i : Y_i$ order.

No! Prize is very large, same for all, but cc_i is cost of best S_i, & prob of win is p_i. Choose in $\frac{p_i}{cc_i}$ order so as to minz expected cost before winning.

→ Different problem: [redacted] best S_i wins Y_i, has prob of p_i of winning: Chose in $p_i : Y_i$ order.

For OZ: S_i has $F(M,T,S_i)$ expected [redacted] G value in time T. (?)

What is $P(M,T,S_i)$ the prob of?

Say one wanted to find the $\tilde{S}$ soln of M,T in "minimum time": This is not necessarily the best way to spend one's time in working on (M,T) or on "M".

In fact: a likely way to work on M, or M,T: pick a S_i we by expected [redacted] per unit time. Use it for a while: As one uses it, the probabilities that one should work on other S_i change. Also one can use info on [redacted] M tries using S_i to help find better M trials for other S_j's.

On the "Optimality" of OZ Lsrch. If $P(M,T,S_i)$ d.f. is "correct" or "optimal", then perhaps OZ Lsrch is perhaps optimal: But, just as in INV Lsrch, $P(M,T,S_i)$ will be non-optimal, & we will be [redacted] "improving it" during the srch. In the case of INV Lsrch, we use OZ Lsrch to [redacted] implement this improvement. Would probly do the same thing in OZ lsrch.

What sort of a priori info do we normally have about M,T,S_i? Perhaps we have an a priori on the [redacted] max value of M obtainable. For each T, we have a diff d.f. presumably, as we work on the problem w. S_i & /o other S_j's, this "error" d.f. over yield, changes. → 16.03

.35

Could I somehow use ideas from the "What to work on Next problem" soln. to decide Which S_i to work on next? That "soln" requires that I have a "prob of success" [redacted] as a func. of time spent" curve for each possl. candidate.

What changes betw. ★ U's Nov 94 & Jan 95 papers? In the later paper, he used some data for the experiment w. TM $\vec{V}$. ||| (He did not mention using "Noisey" input in Nov 94.) Also, he didn't do the "Navigation task". Woops! This ↑ was for the "Navigation Task".

.03: At any time, one has, ■ in view of one's experience thus far, an expected G in Δtime = ■Δ for each of the S_i. Say this is ≡ $G(M, S_i, \Delta)$.

Say the initial problem is M, T. We have now worked on several of the S_i's: ■ used up T_0: we have $T - T_0$ time left. We then chose to work on S_j ⇒ $G(M, S_j, T-T_0)$ is max

We continue to chose S_i's every small ↑ in time; The time we put back jumps depends on how much each jump costs (in time).

The OZ prob is best done in Time Share mode; in which case it becomes an "Anytime Prob" soln.

In a TM, both OZ & INV probs will be worked on either alternately or timeshared: — At any rate, there is only 1 p.d. for both — in the sense that they share definitions, structures etc. — Trouble is, it's not clear as to what the p.d. for the OZ problem is

a P.d. off! Toni, Backlam ②

Simplifying the problem in: Say we have an M, T problem but we assume $P(S_i)$ is a funct. of M, but not T. So $P(M, S_i)$. This will be our estimate of the prob that S_i will give the highest G rate increase; Or $P(M, T, S_i)$ is the prob that S_i would give the highest G in time T.

.21 The easiest way to use this is to assume $P(M, T, S_i)$ doesn't change as one learns more about M & its various "candidates" & their M values.

131.4 km/h = #13/yr.

We'd like to reduce the no. of S_i's of interest. If several are about the same, we want to chose on of the set & give all time share effort to one. (We also want to do this in INV Lrch, but (Maybe) it's not as critical as in OZ)

In Lrch, by ommitting certain cands, we could lose the soln. entirely. In OZ problems, the effect of leaving out certain cand. S_i's is more benign, we get a worse soln, but we do usually get some ↑ in G.

If there are 4 ways to do a H.C. that are ≈ equiv, then putting all wt. into 1 of them gives equiv. ↑ of pcs & ∴ equiv ↓ (by ¼) of soln. time to a given level of ΔG (If that happens to be the solution). ↓ ≡ time share

.35 We could divide up the S_i space into (Quantize) "equiv classes": These equiv classes of S_i's have, for a given T, the same G values within a certain factor. If there are N in the equiv. class, then they need only be equiv. within a factor N (Because we gain N by making the equiv class).

We can put a Ball around one of the S_i. Choose dims of Ball such that ~~Hyast Gimball / Lowest Gimball~~ $\left(\frac{\text{lowest Gimball}}{\text{hyast Gimball}} \times \text{No of } S_i\text{'s in the ball}\right)$ is max. Actually, the "Ball" topology isn't necy. Use any set of subdivisions ∋ $\left(\frac{\text{min in division}}{\text{max in division}} \times \text{no. in division}\right)$ is max.

We want to divide up t. entire space in this way — so that t. Minimum of all the $\left(\frac{\text{Min in division}}{\text{Max in division}} \times \text{no. in division}\right)$ is Max.

.02 This way we are assured of a certain minimal ↑ in pc. of t. actual soln,

---

16.20 – .2 is a reasonable defn of $P(M,T,S_i)$: A good P.d. would assign a very hi p.c. to t. "best" S_i .

How can we use ALP to get estimates of → ?

Actually, I'd like t. p.d. to do t. Entire 16.35 – 17.02 assignment: i.e. pick typical S_i's for each "equiv class" so as to minimize cc for search

⟹ well we would have examples of (M,T,S_i) triplets of t. past: where S_i was t. successful H.C. method. Since only a finite no. of S_i were used in t. past, we'd want TM to extrapolate & create new possl. S_j's & assign pc's to them (wrt M,T).

Stacky Pattern / Optimal Factory / 1.60 2 yrs. / ALEUCO ≃ LLSP.

"Anytime Algos"
In t. "Anytime" OZ prob: we don't know T but perhaps we have a lower bound on it & maybe a d.f. for it.

Looking for a p.d. for $P(M, S_i)$ is easier because, thru "pooling" the T's, we ↑ ssz.

1.14 Getting back to INV Lsrch: From GHT, we know that $\frac{cc_i}{pc_i}$ = constant gives optimum over.

Since at any particular time, we don't know t. cc of any of t. proposed solns, but having worked on some of them for known times, we have lower bounds of these $\frac{cc_i}{pc_i}$. The only possl. reasonable guess for optimum soln., is to keep all $\frac{cc_i}{pc_i}$ constant during t. srch, which is ≡ t. timeshared method which seems to be optimum.

.21

Now: Consider OZ problems: Given M,T to find best S_i . We can regard this as an INV problem with a fixed, same, T for each S_i . Given $P(M,T,S_i)$ we should try S_i's in order of P — each for time, T.

If T is not given, perhaps work on all S_i timeshared with timeshare ∝ $P(M, S_i)$ for S_i.

F Hint, t S_i space would have to be partitioned according to 16.35 – 17.02 !

Just how to do this is not clear: To do t. partitioning we'd need to know much more than just $P(M,T,S_i)$.

.29 According to .14 – .20 Timeshare Lsrch should be optimum; But if many of t. cands are identical, & is also $\sum p(\text{cand}_i) >> 1$, even if all cands are diff, & there are usually >1 solns. In fact $\sum_i p(\text{cand}_i)$ can be < 1 also: $p(\text{cand}_i)$ is just t. proby that cand_i is a soln. All those probys can sum to << 1 or 1 or >> 1. There is no restriction.

.34 If we assign probys via codelength, then $\sum p(\text{cand}_i)$ must be ≤ 1 by Kraft inequality —

So it's questionable whether direct use of coding could give reasonable probys! —

Since t. probys need not be normalized or even Normalizable!

Another difty in INV. Lsrch! That t. proby of soln. of t. various cands. are not indep: If a certain cand. fails in time T, this affect sol'n probys of other cands for time T & for other time Limits.

In 17.34 I was concerned w. t. idea that t. d.f. for Eur L seen was not normalizable: Now this was true if we were concerned w. t. proby that a particular cand. would have (eventually) a soln. — since many difnt strings will solve t. problem — in perhaps all cases.

If (as for the OZ problem) we define P(, x) to be t. proby that x is t. lowest cc soln. to t. problem, the diffty is reduced considerably:

It is not eliminated, hvr, because usually there will be many x's that give identical or near identical cc for soln. — so the ideas of 16.35 – .02 are relevant. → 19.29

:10

— Li–Vit. (first ed) pp 245–246

(SN) On Gac's proof that ALP is within a factor of MDL that → ∞ very slowly as length of seq. → ∞! I suspect that this proof is obtained by using coding methods that are very time consuming! If so, then if we use "reasonable" cB's on our approxns, t. Thm. is not true!

A possi. form for proof of Gac's Thm! A code for strings is of t. form (τ, X).

Consider all output of pgms that take τ or less time: (This is easy to show if we also know how long X is: Do it for X of known length ℓ; then do it for extensions of X of length τ or some function of τ).

So code of S: (τ, ℓ, X): list all X's that give [illegible] same prefix of length ℓ, after time τ. Make equiv. classes of these X's.

For each equiv. class do $\Sigma 2^{-\ell_i}$ (2^("mean length"))]. Order the classes by this parm. — call this order-number "Y". Then Y is the code for that S. (actually (Y, τ) is t. code)

Y will be ≥ the ALP (prect) & yet it will be only 1 less than MDL. If we use a priori, a rapidly ↑ funct of τ instead of τ, we (say Ackermann funct.) instead of τ:

i.e. (Ak(τ), ℓ, X), then for τ = 4 we have a very large no.)

[For practical purposes, use Ak(τ+9), say, so that even for $\tau = \phi$ (t. default value) Ak(τ+9) is enormous.] Ak(3) = 3^{3^3} = 7.6 × 10^{12} = α: Ak(4) is $\alpha^{\alpha^{\cdot^{\cdot^{\alpha}}}}$ α times.

Ak(4) is proby large enuf for any practical considerations! (I'm not sure I'm getting Ak funct. correct — but it really doesn't matter!)

→ Then even for t. default case, (0 or 1 bit extra), we have very little difrnce betw MDL & ALP.

3^{3^3}
$= 3^{27}$
$4^{4^{4^4}} = \alpha$
$\alpha^{\alpha^{\alpha^{\cdot^{\cdot^{\alpha}}}}}$

hvr, note Optical fecta A GO for 2 pairs incl. exam, shocking shocks!

A possi. trouble → t. amount of τ necy to find t. shortest pgm for a string increases non recursively fast, as a funct of t. length of t. string.

Considering "Extension Proby": Lets automatically consider all extensions of length = to corpus. I think that in just about all practical cases, this would be ok.

Not yet! A re-definition of (τ, X): for a given (large) τ (= c.B.), pool together all trunc Inputs that code t. same output. Taking the $\Sigma 2^{-\ell_i}$ of each output for wt. — order t. output types according to $\Sigma 2^{-\ell_i}$, & assign an order no. to each. The most wt. gets 0, etc.

So this is a way to get a code for any string ≥ ... There is only 1 code per string.
Now the Q is about codes w. a particular x (string) as prefix.

My impression of Gac's Thm: That the shortest code that has s as prefix for output, is "not much longer than" than $\ln_2$ (pc of s). Li-Vitanyi PP 245–246 discusses Gac's Thm. & explains how it was proved.

Km ; KM   KM : p 243 $-\lg_2 M(x)$ M(x) universal continuous semimeasure (∴ $\sum 2^{-l_i}$: l_i = length of i'th input w. output having x as prefix.

Km(x) monotone complexity p 243 : length of shortest pgm that produces an "extension" of x as output: i.e. x is a prefix of its output.

(term that is indep of $\ell(x)$.)

The thm is $KM(x) \leq Km(x) < KM(x) + Km(\ell(x)) + O(1)$

If $\ell(x)$ is a random no. (as it usually is) then $Km(\ell(x)) \approx \log_2 \ell(x)$

for a stationary source $KM(x) \approx \ell(x) \cdot \alpha$ (constant). Then $\lg_2 \ell(x)$ is smaller than $\alpha \ell(x)$

— But not very much smaller! So this isn't a v.g. upper bnd. for the difference!

The discn. in Li-V. p 246 on how small the difference betw. Km & KM is, seems quite weak! — But I really don't understand the notation yet — as is; the final conclusion, that Km $(\omega_{1:n})$ - KM $(\omega_{1:n})$ (for almost all infinite ω) is < any unbounded recursive function.

It may be that I'm concerned w. the O(1) term in the difference. — Because it seems clear that in all cases KM(x) is signif'ly < Km(x) because there are lots of strings (other than the one w. shortest code) that have x as prefix.

Drop this for a while. (The 18.20 seems like a good approach to understanding Gac's Thm — and its (possl.) limitations

Anyway, my main concern is w. RLP : The difference betw. $2^{-\ell}$ of the best code thus far; & $\leq 2^{-\ell_i}$ for all codes found thus far. } which may be quite distinct fm Gac's Thm.

.29 : 18.10 This idea of using a single (low precision (hy pc)) point to represent many {hy precision / low pc} points is used in MDL & MML arguments.
Actually, in normal Lsrch (the only kind I've been considering) MDL is the more appropriate measure — Giving them since the individual trial [illegible] ("cand"idates) have MDL proby assignments. — Giving them

.33 ALP proby assignments would ↑ their pc's considerably (due to duplication among other factors), but I haven't been able to discover a way to do it! It remains one of the

.35 impt. outstanding problems.

Getting Back to the General OZ prob: In line w. 17.23, Lsrch for INV probs could be close to optimum: The d.f. could arrange the "pooling" of cands (in 16.35–17.02). This pooling would be done by TM_2 : an OZ problem that assigns probys so that Lsrch will be as short as possl.. The details of how this can be done is not yet clear to me, hvr. But the main idea, is: We want the p.d. for INV Lsrch to be the proby that a given cand. gives the

fastest soln.

There is much confusion in my mind about how hours enter via t. Pid. It would seem that many hours would control each branch in t. search ent. basis of results of trials thus far: By "Results" I mean not only (fit/not fit), but t. details of what went on in t. trials.

We would start out, hwr., with TM$_2$ using only simple uncorrelated trials for each problem. Then later w. more info available, it would consider more complex strategies involving learning during t. soln of a problem. This might mean that t. individual trials took longer, but that t. proby of overall fast soln. would ↑.

—— We might, in fact, retreat to "Uncorrelated trials", but have only a few of those trials per problem (that had much pc). These hy pc. trials would learn during t. trial — they would have many during trial. There only output would be a solution (if they had time to find it).

As time went on, TM may want to reduce t. no. of trials to 10 or 2 — (Maybe just 1), so that it wouldn't lose "between trial" info ——

i.e. normally info is not transferred betw. trials during a run — except via TM$_2$'s changing probys for trial 2 in view of what happened in trial 1 — but usually those mediations would be based mainly on averaging over runs of earlier problems, rather than t. present problem.

It may be that I had a lot about these kinds of problems in 1988, when I wrote t. proposal → Sol(88). —— Perhaps try to find ~TM88 notes. (I think there are some TM 88 Notes —— (may be mainly on TSO) in one/both of stuff I took to SAARB.

Criticism of J's LHL: If we have sudden large bad ΔR, some pop a lot of smp's! This ΔR could be just a random disturbance that doesn't occur often — having little to do w. mean goodness of various smps.

2 Impt Q's: ① Using J's to Gerc: would machine ever be able to realize that t. main idea is long term $\frac{R}{T}$?

② would I use J's system (or mediation of it)? Why or why not?

Re ① My impression: that RTM h = 1 v.s. 100 v.s. "∞" or h = T are all quite different. By not specifying h, I suspect J gets h = 1

Hwr. there is at least 1 other factor that I don't understand clearly:

-38 Say smp 5 has been followed by "useful" smp 6, 7, 8: We then get a bad string of R's, so we pop back to smp 5. At this point, smp 5 has legitimate credit for what it did up to smp 6, but it is also getting ... space 2401

On "Solving POMDP's w. Search & EIRA" call this "W.J." w/ Wang & Schmidhuber.

I only have the first 6 pp of this, but the first 4 pp give the theory. (Thm 3 § 4)

My first impression: That the idea is pretty much the same as LHL paper: That his overall Gov is the same w. its push & pop criteria:

General idea: I think that he's doing very short time-horizon tests of future options. To go further into the future: One way would be to wait much longer before deciding to accept or reject.

J. sort of does this, but he does a lot of "Rejection too soon" — before he has good enough evidence for rejection: If each trial has > 0 overhead, one should tolerate some "drawdown" — size of/ddn dependent on overhead ← Nitro, sec. 2.0

Also, I'm uncertain as to the extent to which his system is able to look at the past & use this info to predict what would be good in the future.

Also, the fact that it doesn't seem to be using info about models, with rapid retrain — seems very wasteful.

A big Q is: Does system has internal storage facilities. If it wanted to, it could remember previous "bad" models: but as it is, it's probly unlikely, because it's so hard to do!

Re: His rejecting emp's on the basis of one bad trial: Well, since it costs ≪ 1000 time steps to prepare a new emp, it would seem best to try a new one if ΔR didn't beat previous average. This was true in LHL paper, (first example): Whether it was true in the W.J. paper . . . I don't know.

In example 1 of LHL: If recents comes infrequently (as in a chess game — only at E. end) one must spend one's time finding things that correlate w/ Recents — to use as (more frequent) secondary recents. — Also, I spend much time trying to find relationship betw. conds & R's.

E B D G
5 2 4 7

A comparision of J's v.s. mine v.s. "Expert systems".

E.S.: Fast to solve probs. but "Brittle": Doesn't learn.

v.s. GPS: more general/less brittle but slow: Doesn't learn.

J's "LHL" More general. Lazarus, but very slow.; very little "Bias": Could be speeded up by putting in more "Bias" a/o by suitable TSQ.

RJS: Learning fairly general (but not perfectly General): Much prior: lots of Bias.
I am trying to get a useful system as fast as possible.

J's criticism of GA is v.g.: They are unable to modify their hyper (own) organization. They could be modified so that different forms of evolution are possible. (As in organic Evolu.) I don't know if people have tried this.

4.12.97 TM: LHL: How to ORDER TRIALS in PC order .01 – .23 ← rather General soln. for other cases as well.
Proof of Gambling House Thrm. 1. .24 – 23.15

.01 : Ordering trials in pc. order: T. main ideas: 5.01–.30, 7.27–.40 ;

T. Problem: I have to chose m probabilities. $q_1 \cdots q_m$.

q_i can be any one of P_{ij} : J can cover any finite or infinite range.

Bickfords 2 for 1 pancake ritz.

I want to make combns. of probys $\prod_{i=1}^{m} P_{ij}$ in pc order: How do I do this?

P12.30

First assume probys P_{ij} $j = 0, 1 \cdots$ are in order of size: P_{i0} largest.

Normalize the P_{ij} so $P'_{ij} \equiv P_{ij}/P_{i0}$: $P'_{i0} = 1$ always.

ordering the P'_{ij}'s is same problem as with P_{ij}'s.

Chose a threshold, T. We will now find all sets of P'_{ij} ∋ $\prod_{i=1}^{m} P'_{ij} \geq T$.

First pick the largest P_{ij} i.e. P_{10}: Next pick P_{2j} ∋ $P_{10} \cdot P_{2j} \geq T$: list all possys

& put them on t. stack: for each possy; next consider P_{3j}: List all possys ∋

$P_{10} \cdot P_{2j} \cdot P_{3k} < T$. & put them on t. stack. essentially a tree search;

In each case we go along t. left hand side going as far as we can

while $\prod_{i=1}^{m} P_{ij} \geq T$.

To get choices so that $\prod_i P_{ij}$ is below T_0 & $T_0 + \Delta$; Go out on t. tree to Thr. $T + \Delta$

Threshold; Then try to go as far along as possbl. w. $\prod P_{ij} \geq T$. ~~T~~ ← 2T

In tL srch, one just wants all selections w. $P >$ a certain T

.23

.24 Gamb. House Thrm 1: That if cost of ith choice is cc_i & proby of i winning is p_i : Then t. ordering of trial w. least expected/ total cost of winning is $\frac{p_i}{cc_i}$ order:

To prove this, suppose that for a certain i & i+1 $\frac{p_i}{cc_i}$ & $\frac{p_{i+1}}{cc_{i+1}}$ are out of order: Then show that by reversing their order, one cay ↓ t. expected cost.

The ~~expected~~ total cost is t. sum of terms: each term is proby that t. ith trial will win mult by t. proby that no previous trials won mult by cc_i.

The sums for i, i+1 order vs. sums for i+1, i order differ in only 2 terms. T. terms for i, i+1 order are

.30 $$\left(\prod_{j=1}^{i-1} (1-p_j)\right) \cdot p_i \cdot cc_i + \left(\prod_{j=1}^{i-1} (1-p_j)\right) \cdot (1-p_i)\, p_{i+1} \cdot cc_{i+1}$$

T. terms for i+1, i order are — errors! See 23.01

.36 $$\left(\prod_{j=1}^{i-1} (1-p_j)\right) \cdot p_{i+1}\, cc_{i+1} + \left(\prod_{j=1}^{i-1} (1-p_j)\right) (1-p_{i+1})\, p_i\, cc_i .$$

This ∏ cancels out so it's $p_i\, cc_i + (1-p_i)\, p_{i+1}\, cc_{i+1}$

v.s. $p_{i+1}\, cc_{i+1} + (1-p_{i+1})\, p_i\, cc_i$

subtracting out $\left(p_{i+1} cc_{i+1} + p_i c_i\right)$ from both: $- p_i\, p_{i+1}\, c_{i+1}$

v.s. $- p_{i+1}\, p_i\, cc_i$

So t. ordering depends only on rel. size of cc_i & cc_{i+1} : clearly I made an error! → See 23.01

~~cost limit~~ U w.

4·12·47 TM : LHL

.01 : An error is (at least) in 22.34, .36 : ~~The corrected note is~~

In 22.34 : "cc_i" should be replaced by $\sum_{j=1}^{i-1} cc_j + cc_i \equiv A + cc_i$

cc_{i+1} " " " " " $\sum_{j=1}^{i-1} cc_j + cc_i + cc_{i+1} = A + cc_i + cc_{i+1}$

.04

$$p_i (A + cc_i) + (1 - p_i) p_{i+1} (A + cc_i + cc_{i+1})$$

$$p_{i+1} (A + cc_{i+1}) + (1 - p_{i+1}) p_i (A + cc_i + cc_{i+1})$$

Certain terms cancel : i.e. $- p_i p_{i+1} (A + cc_i + cc_{i+1})$

$p_i (A + cc_i) + p_{i+1} (A + cc_i + cc_{i+1})$ The A's cancel out :

$p_{i+1} (A + cc_{i+1}) + p_i (A + cc_i + cc_{i+1})$

$p_i cc_i + p_{i+1} (cc_i + cc_{i+1})$ so $\left(\frac{p_{i+1} cc_{i+1}}{p_i cc_{i+1}} \right)$

$p_{i+1} cc_{i+1} + p_i (cc_i + cc_{i+1})$ v.s.

So the relative sizes depend on whether $\frac{p_i}{cc_i} < \text{or} > \text{or} = \frac{p_{i+1}}{cc_{i+1}}$

.05 So that proves it !

·01: Spec 20.40! "credit" for t. $\dot{R}$ during smp6,7,8 · looks 2.24 & 12.30; In t. long run, hvr., this finite bias in credit, will become unimportant. — It does, hvr, give us much more "drawdown tolerance" for future testing of smp5.

N.B.: That in addition to t. extra ΔG given by smp6,7,8; t. contents of t. "workspace" has changed (betw. start of smp6 & time of [redacted] beginning of "pops". The meaning of a $\vec{P}$ vector: ([redacted] ≡ t. $\vec{\vec{M}}_{ij}$ matrix) depends on t. contents of t. workspace to some extent.

To some extent, when one pops at end of smp8, down to smp5, one loses the R info that occurred in that time. It is applied as a bias to t. evaluation of smp5, — but in a direct way, it is an "undeservd" bias. ≢ In a more general sense, t. bias is deserved because we now know that smp5 is capable of being improved / in a short run by at least t. amounts smp6,7,8 improved it.

The increase in $\dot{R}$ during smp6,7,8. is "remembered" as t. modifn. of t. contents of workspace during t. testing of those smp's. Whether smp5 can ~~[illegible]~~ exploit this info in any way, is unclear. → (.3·4)
(it also remembers t. last ΔR: see (.20))

·17

·20: **SN** Well, I found it! LHL: 13.30! The latest ΔR is written into an input cell. They all $\vec{I} \rightarrow \vec{\phi}$. So it does have access to t. last ΔR; also IR, SP and t mod 10k.
It also can know what in $\vec{V}$ ≥ write to $\vec{V}$ as well.

There is definite confusion about addressing! T. reg. area seems to be $\pm\left\lfloor\frac{nops-1}{2}\right\rfloor$
But addresses are all from (0 to nops): This gets into t. pgm area! ~~The~~
~~[illegible]~~ J says (6.10-.15) a gt pts as cells in reg area for most insts (apparently not all) Well, ~~[illegible]~~ All Jmp insts. prob'ly point to cells in pgm area.
So if all other insts point to cells in reg area, how do we get out of t. reg area at all?
Well t. machine starts w. small nos. in t. reg area, but can do arithmetic on them, so t. nos. in ~~t.~~ reg area can be very large or small. Since these nos. are never "reset", it seems likely that eventually one would have randomly large nos. in reg. area, ~~[illegible]~~ so they could then refer to arby storage cells. In fact, this may be what occurs: we eventually have t. reg. filld w. random nos betw. — 1000 & +9. These point to other cells that ~~[illegible]~~ can store info.

·32 Since there are only nops ~~[illegible]~~ = 19 difrnt addresses possb, there are only
·33 19 difrnt possl. places to address. What's t. point of using indirect addressing? → (26.27)
└ a peculiarity of "C"? — seems unlikely!
→ a possl. point

·34 (.17) → Well, ~~[illegible]~~ Ignoring memory of workspace content, t. machine climbs in $\vec{\vec{M}}_{ij}$ space.
It tries to find point w. max $\frac{\Delta R}{\Delta \delta}$ (since ~~[illegible]~~ last modifn.) If $\frac{\Delta R}{\Delta t}$ drops below that of t. previous smp it pops to that smp (→ $\vec{M}$ value) [The smp is t. step: ~~[illegible]~~
action of
The original (uniform) $\vec{\vec{M}}$ plus / all t. useful smp's give present ~~[illegible]~~ $\vec{\vec{M}}$.]

From this perspective, we tend to get a $\vec{\vec{M}}$ that has a large as possl. $\dot{R}$ for as long a time as possl. (double optimization! ☺).

One can't really do a partial ordering on T, Ṙ pairs: It doesn't work that way. We do tend to get large Ṙ for large T, hvr.

T

Ṙ

.04

See 20.38 - 40; 26.01 ff for discn. of this & other aspects of the problem.

I see a possl "equilibrium" state, in which we have a local optimum, $\vec{M}_{00}$

We make models of it that seem to work for a while, but eventually they don't & we pop back to $\vec{M}_{00}$ & try another samp to modify it.

T. local optimum of .04 will perhaps be very much dependant on the initially random search of $\vec{M}$ space.

→ Now .04 would probly be true, except parts info in workspace ("work" & storage, can input registers) that is not initialized after each "pop". So the state of the systems ($\vec{M}$, $\vec{W}$), ($\vec{W}$ is contents of workspace) we will climb on $\vec{M}$, but $\vec{W}$ is not nearly changed in/response to motion in $\vec{M}$ space.

What may happen: We get at a local optimum wrt. $\vec{M}$ → $\vec{M}_1$. We continue to try various $\vec{M}$ values near $\vec{M}_1$; we also have various changes in $\vec{W}$, so even if $\vec{M}_1$ is a local optimum, changes in $\vec{W}$ could move it to an effectively different point — possibly better — but I don't see any hill climbing activity wrt. $\vec{W}$ — $\vec{W}$ seems more or less "randomly" changed, if at all! If we find a V.G $\vec{M}$ then eventually, the $\vec{W}$ assoc. w. it will change, & that $\vec{M}$ will no longer be so good!

— A possl. way out would be to obtain a $\vec{M}$ whose effect is indep of $\vec{W}$, or have an $\vec{M}$ that makes ~~is of very small change of the effect~~ the effective change of $\vec{W}$ extremely unlikely.

Suppose we evolve a good w. We would then like to make it unlikely that any write instrs are given to change it. Init, Mov, Dec, Inc ... In fact all instrs. but the Jump instructions can write to a (≈ nearby) cell. → hvr. see 26.01 → 26.01

.24

What may conceivably happen: The system finds a M, W pair so that while $\vec{W}$ is rearranged often randomly changed, the average effect on the ppm is minimal.

Note that M denotes a stochastic ptm, & $\vec{W}$ can be considered a kind of stochastic Variable.

T. hoopp. is relevant if the input problems stay about the same, so there would, in theory, be an optimum point for $\vec{M}$, $\vec{W}$. (but $\vec{W}$ will drift, & it can drift arbitrary amounts)

If, in addition, the nature of the input problem is changing, then the analysis becomes even more

---

Since $\vec{M}$ is the adaptive part & $\vec{W}$ the "drifty" part; it would be OK if $\vec{M}$ could adapt to $\vec{W}$'s drifting. (As is, it's difficult enuf, even if most of $\vec{W}$ were stable, that the input problem part of $\vec{W}$ would drift — an essential part of the system that we want $\vec{M}$ to be able to adapt to). What it looks like, is that the non-input part of $\vec{W}$ seems to add to the problem.

→ spec 28.20

Also 156.01

.01:25.24 Looking at t. insts aggm: Many, if not most of t. cells will be occupied by arguments of insts. These can be specified by $\vec{M}$. The cells they refer to can remain fairly constant over time. For any useful pgm to execute, certain of t. cells refered to have to be identical. There are only 19 cells, so this identity is possl. but not as likely as it should be. I think t. problem may be that J has devised a lang. that is not so good for this sort of thing.

APL was designed so that a random pgm tends to be meaningful — even interesting.

Using J's inst. set, too few pgms are meaningful (or even syntactically acceptable)

e.g. Add (a,b,c) Mul(d,e,f) cannot interact unless either c=d or c=e.

→ Using a lang. in which all or almost all random strings are meaning for a/o interesting would be a big advantage

.10 Another point: Init(a,b) is probly unnecy. Inc & Dec are able to "seed" t. system. Say we start w. zeros in all but the pgm section. I'm assuming Reg. addresses from 0 to nops, pgm starts at nops+1 → 100; All addresses point to cell 0. It is eventually incremented or decremented so it pts to cell +1 or cell −1. Move (ϕ, a) will move ±1 to a.

The first Major Breakthrus of Koza's was t. fact that his trials had a much higher fraction of meaningful, useful cands, than ordinary crossover/mutation techniques. T. main advantage of J's lang. is that probably t. insts are identical to "C" insts., so its easy and fast and "UNBIASED"!

.20 Re: Indirect addressing: I don't see t. pt. of this! He only has access to 19 Many cells, whether they are directly or indirectly addressd. (Maybe a peculiarity of "C"?). ← same as remark 24.32 24.32

That method I had of Generating all functions via a tree, had no meaningless trivial functions.

.27 ; 24.33 AH! The registers do sometimes point to the pgm area, so defintly indirect address is difrnt from direct..... but exactly how? He has not direct access to cells (other than t. unnecy "init" instruction.. (.10)) — Hvr, the content of t. reg. cells can be changed by indirect access, so we have sequencial access to perhaps all of RAM! e.g. say reg cell "a" has ϕ init.

:32 To add t. contents of cells 1 to 10000 Inc b ("b points to a") ; add b, c, c : Jump if b content < 10000

still, do we need > 19 + 100 addresses? → Hvr., see 27.07 for a (telling) agt against this! → 27.07

(LHL)(J) What does it mean for an address to point to a cell in pgm area? Can t. cell content be chngd?
6.13
6.14 says: reg cells point to cells in Storage: "storage" = Min to Max (Also see 4.39 on what "storage" is.)
6.25–.27 "Read" insts may point only to cells in "storage"
Write " " " " " " (Min, pgmstart−1) (not pgm area)

What this implies: If a "read" inst. points to pgm cell that hasn't been assign yet: unclear as to what to do!

Each cell content depends on content of cells leading to it (i.e. to decide whether an inst. no. or a...) No! The assignment of an integer $-1 < n < nops-1$, is indep of how that integer is to be interpreted,

.01 If a pgm call hasn't been "filled" yet, then we can fill it from $\vec{M}$ whenever its value is required. Actually, one could assume that the system acts as if the entire pgm space had been filled by $\vec{M}$ before each trial. This is, here, quite wasteful of time. . . . But one could run the system as if it were done that way, creating pgm calls only when needed. This would allow writing to pgm calls, but J apparently chooses not to allow this, & allows reading only from those calls.

.07 Something wrong w. my thinking! If we don't allow read or write from pgm calls, then it would seem that it's equiv. to only having "nops" diff. storage calls for direct addressing! Yet .07 allows the pgm at 28.32, which clearly uses >> 19 calls.

Well, the fault in reasoning is that while at one instant, the 19 regs refer to only 19 other calls — the "19" can be changed arbly. Like in DOS, expanded or extended memory, we have this 64K "window" that can be put anywhere in 32M of RAM.

What it means is that one has access to a very large RAM, but it takes a little longer to get to the distant places! One must increment addresses first.

Perhaps a better way would have a certain amt. of RAM for direct access, & a much larger part available for indirect access at lower cost ... — e.g. "in line" data would be nicer; if we give it $pc = 1-\epsilon$, it could be very stable; say those "pgm calls" could not be written to by other instr., so they really could be fairly stable. Would like to have whole sets or data tables be "stable".
We need ≈ "Memory Protection".

What J's system is, is like a large time sharing system at a University, in which all students are "Super users" & can read & write anywhere, anytime.

A way to deal w. this: have probs values of 2^{-N} & $1-2^{-N}$ available, with the probability of "N" & w. N, but not too rapidly.
Or, just have probs of 2^{-N} to 2^{-M} being assigned rarely instead of often, but $\sim 2^{-N}$. This could mean that we can write probs as a N bit binary fraction for an arby N, but N bits of precision have $pc \sim 2^{-N}$. So .1111 = $1-2^{-4}$ has $pc \sim 2^{-4}$!

Another way to get something like this would be to use J's system but not have a lower bound on pc's: — So we could create arbly small or large pc's, but never = 0 & never 1.
This would make it not so easy for J's "Get P" instruction, since P could have arby precision!
Well P could be in floating pt w. 1 or 2 or 3 bit mantissa & only a max. possible exponent.
Actually, this wouldn't work: We need hy precision for pc's near 1 as well as near 0.
So: probs's first digit is "is $pc > \text{or} < \frac{1}{2}$?", whether if $< \frac{1}{2}$ we give 2 or 3 bits for exponent.
If $> \frac{1}{2}$ we look at $1-p$ and give 2 or 3 bits for exponent.
We need hy precision only when P is near 1 or near 0.

For IncP & DecP may be only have 1 or 2 or 3 possible factors of ↑ or ↓.

J is using MinP = .001 This is the smallest value a prob can have. Since probs are normed, no prob can be $> 1-.001$.

I may be dealing w. 2 kinds of probability. One is the prob assoc. w. $\vec{M}$ elements;

■ The direct effect of/or probys near 1 i near ϕ is small.

Another kind of probly, hvr, is assoc. w. elements of t. storage cells (≡ prgm cells):
We want certain of these cells to be very stable, so we can store good, useful ideas, etc, w.o. having them being overwritten. (Perhaps) infinitely, t. M probys make it possl. to write into or by storage cells, so if the M values have precision ϵ, there is a proby of $n\epsilon$ of overwriting any storage cell.
One patsy would be to have a lit of write once memy; someone could use it to store params one really wanted to save. Q: We'd have to make it hard to use, to prevent a pgm from simply filling all of it w. ϕ's!

Re: Modifn of Probys: Not Inc P or Dec P! Move P toward $\frac{1}{2}$ or away from $\frac{1}{2}$. No: we may want to change a proby from .25 to .75. (from $< \frac{1}{2}$ to $> \frac{1}{2}$).
→ T. most form of algm for changing pc's is unclear

So essentially 2 (probably closely related) problems: ① what form of probys for elements: precision, hy limit low limit. ② How to make long term storage possl. & useful & stable.

.20: (25.40): Moreover, T. slowly changing problem downwt. "input" calls constitute a trgt. TM is supposed to adapt to it. ("TRACK" it). Conceivably, TM could learn to adapt to the random changes in rest of storage. One difference: Some differences: ① rest of storage is large, >> "input area" ② The R input is directly related to t. rest of t. input vars. It is (ordinarily) not related to t. contents of "t. rest of storage". Hvr. "rest of storage" could mirror parts of "input": It could store ΔR data, etc. → 156.01

.25

(SN) Q: what's this about "writing on t. walls"? → 6.37 (LHL) The system has no other place to write! Well, he considers "envt." to be "input & output area": So writing int. "envt" would be "writing on t. walls"

■ J's paper: "A Gen Method for incremental self improvement and multi agent (esp. in unrestricted envts." [" A G M " A Gen Meth:]
Dec
In ¶ 1 of paper (also in t. Abstract), he explains what he means by "Identical trials". I.e. for same input at a later time, t. same soln. as given earlier, would get same ΔR. He says (rightly) that this is a not realistic setting for R.W. problems.
(7.257) → Hvr, In t. human's brain (a self-regulated interval envt.) repeated identical trials are possl., & this is probly where most hill climbing occurs.

4·14·97 TM: LHL: **RTM$_h$** effect of h (HORIZEN) on behavior of TM

.01 On **RTM:** Effect of h: I assumed that as h ↑, the fraction of time spent on long term system improvement would ↑ so that at very hy h, very little time would be spent on immediate problems. Now, it seems this may be false. That even for large h, a certain large fraction of time will be spent on immediate problems, because ultimately, that's what the payoff is based on!

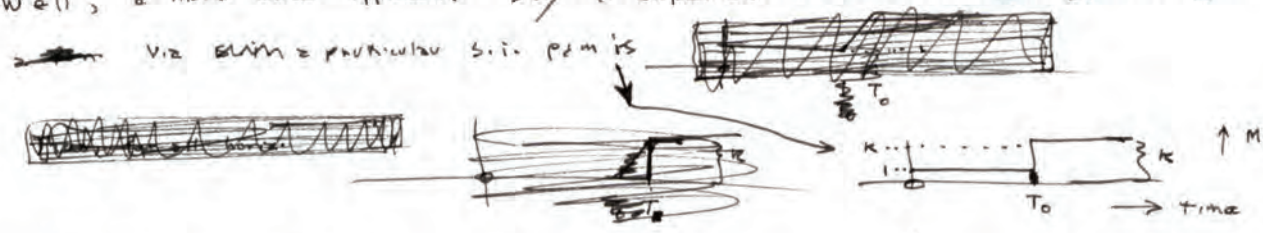
As h ↑, the qualitative kinds of long term self-improvement projects that will be considered, changes. ... But if, on the average, I simply spend less & less time on immediate problems, it's only because I expect the S.I. (self-improvement) efforts to payoff so that I get even more output for those small expenditures on immediate problems. There is the risky, that with larger h, TM will not even start working on immediate problems until ~ the time h has passed — in which case $h \to \infty$ could be catastrophic!

.04 For a TM having a pool of problems to choose from: It has the choice of first spending T immediately on S.I. than working on problem A; or working on problem A first than spending T on S.I.: usually, doing S.I. first is better, because it usually means it will cost less to solve A, so total cc is less. Say it has spent T on S.I. Should it work on A? Well probably not, because if it spent another T on S.I. than doing A would be even cheaper!

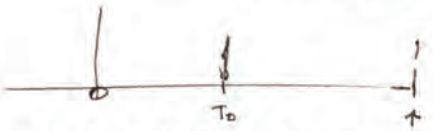
Hvr, say TM has lots of problems in the pool. After doing S.I. the first T, it could do $A_1, A_2 \cdots A_{100}$ rather rapidly & get lots of ΔR. If this would take time T, than if its horizon is $\ll T$, than working on S.I. would not help solve cost of $A_1 \cdots A_{100}$.

1996 95 94 93 Bad
ok 92 91 ok
ok 90 ok
1989 ok 8 7 6 5

.24 Well, a more exact approach: say the expected multiplicative $\frac{\Delta R}{\Delta T} \equiv \dot{R}$ viz gain from a particular S.I. pgm is



For horizons $> T_0$ say T is horizon $T_0 < T$.



Than it is always better to do the S.I. first, than do the tasks.

If $T_0 > T$ than one shouldn't do the S.I. at all.

Hvr, an alternative, rather than doing S.I. for T_0 time than do regular tasks:

do regular tasks for entire time T. Say one's normal $\dot{R}$ is $\dot{R}_0$:

total $\Delta R = T \cdot \dot{R}_0$ if we don't do S.I.

If we do S.I. for T_0 first, than do regular tasks for time: $T - T_0$!

.40 $\Delta R = (T - T_0) K \cdot \dot{R}_0$; so doing S.I. (1 thing first) is better if $\frac{(T - T_0) \cdot K}{T} > 1$ | $K > \frac{T}{T - T_0}$

$1 - \frac{\tau_0}{\tau} > \frac{1}{k}$; or $\frac{\tau_0}{\tau} < 1 - \frac{1}{k}$ so if $\frac{\tau_0}{\tau}$ is small & k is large, then do s.i.

$-\frac{\tau_0}{\tau} > \frac{1}{k} - 1$

if $k = 1.1$ $\quad 1 - \frac{1}{k} = .0909 \approx .091$,

so ~~[illegible]~~ τ_0 has to be $< .091\,\tau$ before s.i. is feasible.

However if $k = 2$, τ_0 ~~[illegible]~~ need be only $< .5\,\tau$ for s.i. to be best.

If τ is very large, then ~~[illegible]~~ there are very many low k s.i. methods one can do, that have a total τ_0 which gives very small $\frac{\tau_0}{\tau}$ so

$k > \frac{\tau}{\tau - \tau_0} = \frac{1}{1 - \frac{\tau_0}{\tau}} \approx 1 + \frac{\tau_0}{\tau}$

i.e. $k > 1 + \frac{\tau_0}{\tau}$ is an easy criterion for a s.i. plan to meet.

$k - 1 > \frac{\tau - \tau + \tau_0}{\tau - \tau_0} = \frac{\tau_0}{\tau - \tau_0} = \frac{1}{\frac{\tau}{\tau_0} - 1} \approx \frac{\tau_0}{\tau}$

Note: working problems gives info useful for s.i.

IMPT

so as $\tau \to \infty$ one might wall delay to $\simeq \infty$ doing any real problems. — Self Improvement

A possible Counter Argument: Doing actual problems gives info that is useful for s.i. However, doing partial problems (not nearly finishing them) would yield about as much info. — The motivation to finish them would be there, however. If one did 90% of a problem just to get info, one would get 10 times normal yield per unit time by finishing it. On the other hand it may be that later, after s.i., one could work the whole problem 20 times as fast — so it would be better to do s.i. first, in that case.

more Criticism 146.01

help!

On inability of J's system to do "experiments": If it has done a new sup in its $\frac{Q}{Q\text{ prime}}$ is k, then after time T, it will be able afford losses of $(k-1)T$ w.o. being poped. After the machine has been running a long time, k is close to 1 ($k-1$) is very small, & we have to wait a long time to try experiments of any size!

hardware Internet connection for

On the other hand, if TM knows it's doing an experiment & it would seem reasonable to risk experimental losses (to gain info) of size $> (k-1)T$.

So essentially the system can do "experiments" only if it "ahead" enuf to warrant an "experiment" of a certain size.

My Model of Induction Using "crumby" CPU's is similar to J's LHL system:

"system C" "system J"

In C, the output depends on ① the input string (which is known) & the particular random interpretation of that string by a stochastic machine, or unknown stochastic process. To get a prediction we take the known string as constant & we feed it (repeatedly) (w. random contrary) into the stochastic machine & look at the different outputs.

In J, we have a known stochastic machine, w. a certain non-stochastic input ((the external + internal) envt. — The internal envt. is determined by the history of the trials of the stochastic parts.

Differences betw. Me. & J. on Goals, Methods.

J. is interested in a TM able to learn real time, realistic input sequences w. only reenforcement vacuity function (not nearly very frequent ΔR). Its goal is understanding how learning can occur ~~in~~ by humans/animals in realistic envt. That ~~[redacted]~~ example problems could be chosen to be, noiseless, clearly separated into sub-problems, that recognit. should be noiseless & always t. same for same problem, be regarded as unrealistic. Developed a

While I feel that these goals are fine, my own goal is to get a TM able to learn very diff. problems; smart enough to read & understand human text. I feel that giving machines lots of certain kinds of help, hints, hours, sequenced presentation of probs, division of envt. into parts, preQuantizing, ect, can be v. useful — speeding up learning tremendously — but they have to be done intelligently — or one could get behavior that looks intelligent, but isn't. I'm confident enough of my understanding of things, so that I don't much worry about ~~that~~ spoon feeding TM a lot.

Also, there are certain "adequate" probs in learning, s.t. if one had a machine that could solve that class, it could solve any other (much more general) class of problems — that second much more general.

→ Actually, my main goal is to construct a very useful TOOL — a very smart machine that I can get to successfully work on many diff. problems at my choosing.

J's all over problem is much more general than t. prob. I'm working on. His technique admits of a much larger class of solns. than mine does. However, because of his generality, t. probly of his system finding a v.g. way to learn in ~~an~~ w. acceptable cc & acceptable t.s., is small.

4.20.97 TM LHL "Crumby" "Crumby"

.01! 30,40': Note: in SystemC I have to have some way to "Reset" the CPU's, so its outputs start nicely indep. of the previous trial. So the randomness in these CPU's, should/will have memory.

{ Is this a new card? . . . .

I have been thinking of "crumby" cpu's as being unfleshed chips: However, if the rest of QM computers, it might be more interesting, & more like what this analysis will be used for.

Using Crumby chips: The reset condition could be a power down condition — in which the cross library part of the chip has memory of its previous run.

In such a trial, we could have a fixed envt. (≡ "input") plus a part of memory that the cpu can use for changeable storage.

I also want "sequential property" machines (monotone, process).

SN It might be possl. to get a simple "cpu" to do ≈ ALP by trying to code 10 or 15 bit streaches at a time, of a lazy corpus. SM data is possl. i.e. use 3 bit precision of Δ prizes. 9 bits is 3 values. At all times, retain best, say 10 or 100 codes. Back tracking (≈ resetting) is back to a point at which the bitcost/corpus length for that seq. was not so bad.

I could try, if not w: $S_n = aS_{n-1} + bS_{n-2} + \text{noise}$. 2 corpus.

A B.z Q is: this "Back tracking" Q!

each pt in corpus has a pt. in code sequence at which it was pointed. When we get a non-match between output & corpus, we go back 10 bits say.

Anyway, this Q of just how to do reset must be looked into Much!

Say we did a ≈ LSearch, but w/ cc $Pf(\ell)$ ℓ is how far in the corpus we've gotten.

Say $\log_2(P \cdot 2^\ell)$ = no. of bits saved thus far.

w/ $P_c \cdot 2^\ell = 2^{\ell - \text{boost}}$. So if a method got ahead, it would get more w/ & more opportunity to get ahead — ≈ positive feedback → "climbing"; So once you get enough ahead, you tend to get ≈ all of w/.

5/15/97 TM

On t. General Path problem: An Approach:
Use t. standard protocol for problem solving:

1) describe t. problem in an exact way:
2) what as some conditions?
3) Can we break main prob. into sub. probs
4) etc. see main Reference.

3rd Germany $3/mo
12¢ England $3 unopened for first 6 mo.
6 mo
1 800 222 0300
ATT 1 flate
in + plan.

Shift print screen
doesn't work in wind → Dos.

Actually, the stuff in "Path" was ≈ first approach: It gave lots of approaches to t. problem. — So t. problem becomes, organizing t. approaches in a conceptually accessable way (≈ Outline, hypertext, various tree strs, graph w. lines relating related approaches).

One Good sounding way, is to work backward.
A main goal is learning to read English in various domains. One approach would be having TM first learn t. concepts in an Algebra text: leaving out problems tied up w. understanding of R.W.
Then we have TM learn to understand t. text that describes what it knows.

Then we give TM Text describing more advanced Math concepts.

Another approach: We somehow get TM to understand some domain & able to read about it. We then Give TM text mainly about that domain, but w. other stuff that it has to try to understand. If TM has hy and E.P.C. it will be able to learn at a reasonable rate w.o. guidance.

Kurt, we can have 2 kinds of input for TM. — One is more like a T.S.Q. By a teacher that understands how TM works. Another kind of input is less carefully structured. TM can have access to a dictionary and encyclopedia.

In addition, we can have several peoples/groups, working on inputs to TM that are ≈ TSQ's: By having these many Domains in which is sort of understands, it will have better chances to make sense of what it reads in t. "open literature".

Could I somehow use Lenats "Cyc" as a useful "input" or "interaction mode"? Perhaps read up on "Cyc". — Try to talk to Ken Haas.

Children's stories have small vocabs but supposedly involve very diff/t assignment of correct nouns to/pronouns that refer to them. This may be "diff/t" because of

a poor lrng. algm. or maybe an inherently diffl. problem.

Another potentially rich pool of input: Take existing expert systems and devise TSQ's for them. We end up w. very useful, non-brittle, E.S.'s that can make probable judgements & learn in a v.g. way.

Each of these Training Envts. discussed, could be fed to TM in (1 ~~case~~) a/o sequencially

So, It would be great if we had a common format, so that I work in many areas of learning ← we can take their TSQ's & feed them to TM!

Well, even w.o. the common format — can we take a lot of stuff in current ML rsrch & put it into a form that TM can usefl take advantage of.?

A common thread, of course, is Alg. Compression — so if any lrng is any good, it can be expressed as compressn & perhaps be xlated into a common format.

[T hyp. is a new version of the Kolmogorov-Chaitin "approach to A.I. (Dartmouth 1956)] (3)

A Diffy w.t. hyp: Much Lrng Rsch requires that the learner start out w. certain given cncpts — So in general, a certain chunk of ML rsrch. would best be given to TM when he had acquired those needed cncpts.

(SN) Prob'ly I should try to get TM to solve OZ probs by learning: I can give it its OZ problem solver(s) until it's smart enuf to start devising new ones on its own. However, I may make it possl. for TM to improve my OZ machine — say by, using SGA, to make better languages or improve the old ones. Perhaps the lang used in SGA could be somehow related to TM's "main P.D." (i.e. its "knowledge Repository").

Some other sources of Trng for TM: If TM has existing Math proofs; filts them in w. parts not given, then, it tries to understand the proof by trying to find ways it could have been discovered. After it's done this, it's close to being as good as having discovd the proof itself. (as close as it can get — at reasonable cc!). — So it will have acquired very useful cncpts. — so that using them it "could have discovd" the proof at acceptable cc.

So maybe the idea is this: At each point in TM's life, he is able to usefully accept certain tcq's or "Training Envts". Each "Life pt", "Trng envt" pair will have an associated cc of putting TM into condition (for trng envt. into condition) so that TM can usefully accept the trng envt.

(useful advantage of)

spac
35.20

.0 (SN) on the uncreatable 02 problem! To find string $x \Rightarrow M(x) \cdot f(t)$ is min. (read no) (t = elapsed time at which soln is given; $M(\cdot)$ & $f(t)$ are known to TM. (3)

.02

.03 Say $\exists$ algm $A(M(\cdot), f(\cdot)) \to x$ that does this & for a broad class of M's, f's:

or $M \equiv M(\alpha, x)$; $f = f(\beta, x)$: α & β are strings that describe M & f.

so $A(\alpha, \beta) \to x$: Well, if A can look at M & f, then it knows what it does.

— Putting it in α, β form adds nothing! A & f can be unrelated.

Well, let's see! : we can have one big M that looks at our algm. inputs $M, f \to x$. It's value depends both on $M(x)$ and on $f(t)$. (i.e. $M(x) \cdot f(t)$ is for any other function of x & t. So does this related .01–.02 to ordinary 02 problem?

(SSN) T. reason this 3 pass is imp't. ! I think it arrives in tournament chess, & IM TM_2's Gore (i.e. t. defn. of TM_2's Gore)

A possl. reason why we haven't been able to solve it: That t. criterion for soln. is not appropriate. I.e. defn. of soln.

Some poor defns of soln: ① That we can get a soln that is slower than that of $A(\ )$ $(.03)^3$ by a factor $pc(A(\cdot))$ ·(2)

.20 (spec 34.40) : So some BIG Q's: ① Look at some current "Tuy Envts" in t. Literature. How can I put them into a form so that TM can use them? (2) Just what state of intelligence must TM be at to accept this Tuy Envt? (3) How much work must I do to get TM to usefully accept this Tuy Envt? (4) How common are useful Tuy envts. in t. ML (literature?

Some currently available Tuy envts: (Some require very hy level of TM!).

1) Text books: These can be worked over to get segus of problems in acceptable order. W. Swartout T.M., we need to do less of this. We can use "hints", suggestion channel, modifns of problems, to make them easier for TM to solve.

RIASEI

4 30 / 45 — 530-6P

2) CYC: Unclear as to if & How this can be used.

3) Lenat's AM: (similar to (3) but maybe easier.)

4) Just reading printed text in general & trying to compress it (i.e. find regys in it). Consider various kinds of text: which would be easiest; a/o scientific at lowest level of TM sophistication. Maybe read "Blocks World" conversations!

5) SM data: (i.e., I don't see much synergy — except w. other kinds of prediction problems. (which last could be as general as "all learning"!)

T. system I have in Mind! Consistent "P.D.": This is to ~~be~~ rapid, & Language, & Bias that TM has at a particular time.

.03 There is a necy ambiguity in Defining at what TM's "P.D." is at any particular time — For many (if not most) Pd's, t. actual probs values one obtains, dependent t. cc. available. A given "P.D." will be t. instrument of creation of [redacted] RLP (Resource Limited Probs)

.06 Using this modified defn. of a P_d. (1) Can we have useful versions of Lsrch?

.07 (2) The meaning of "Improving t. P.D." becomes even less clear. (Tho I'm not sure it was ever very clear!).

.09 for .07 One good approach (for my understanding: to help solve it & to help others understand it.)

.10 Give various examples of things that certainly do "improve t. P.D.".

SN TM uses t. p.d. for Lsrch to solve [redacted] Various kinds of problems.

i.e. (1) Inv. probs (2) finding patterns in a corpus: inductive inference: finding short codes for a corpus.

(3) OZ probs of various kinds: of which (2) are common types.

Note that Early TM will not use t. P.D. for soln. of OZ probs. in t. usual way. It may solve some OZ probs by conversion to Inv probs (a very non-optimum soln. method)

Early TM will ~~use~~ Optan solve OZ probs using optan techniques that I & others have derived. In SGA, we do have a p.d. (stock lang) which we successively improve — but I don't know (if/how) we can usefully identify this p.d. w. TM's main p.d.

.22 Ways to "improve" P.D. (1) Remembering solns to problems: "Near by" problems can be more easily solved, since their solns tend to be algorithmically "close" Near by ↑ search kind.

.24 [redacted] While this ~~is~~ clearly improves TM it's not clear just how t. p.d. is "improved": → 37.01-.04 (Case Based Reasoning)

Actually, I'm not really interested in nearly improving t. p.d. (.22) is an example of TM, being improved — presumably by TM, .

An example of p.d. "improvement": T. P.D. is improved because it assigns a ~~shorter~~ higher pc to t. known corpus in ℰ t. same cc as before. P.D.'s then can be partially ordered w.r.t. p.c. & cc that they obtain per known corpus.

Basically, we always retain t. same Recursive aprip for cc = ∞ & initial corpus: but as time goes by, t. corpus ↑ & our p.d. for t. corpus at various cc levels changes. At first, t. best code we find for t. corpus is simply Bernoulli: — Then we find simple Markov codes, Then maybe finite state probabilistic codes, etc. — each codes t. corpus better & is obtainable at some cc less cc than using previous codes (?).

.38 ~~It~~ My mind is not at all clear on ~~the~~ t. Q. of under what conds one partition (cc, p.d.) set is better or worse than another (cc, p.d.) set

CASE BASED REASONING (.01–.04)

Z141 .11

.01 T. system in Solby (Israel paper): T. system was improved when it solved a problem, because t. code for t. soln contained sub-trees of interest. [If we had a similar problem in t. future, we could use "1 shot" learning" to get a quick soln. for it. [" Case based Reasoning "] — Note 36.22] — Not so easy! The common sub-trees are only recognizable when t. (atom) formal corpus is included.!

.04 T. Solby system improved itself by inventing new definitions. — But t. "1 shot ling" of .01 – .04 could also be impln. In 1 shot ling, we have only one case thus far, so we can't yet devise a defn. T. defn occurs when ~~t. predn~~ one of t. possl. continuations of t. corpus increases t. ssz of t. defn. to 2. → 38.06

.10

.11 This ↑ seems relevant to Z141 (whether ~~t.~~ rite reptn no.? or whatever?). Also that stuff that Riss. did on "content" Z141 is perhaps relevant (or ~~is~~ More exactly, Z141 is relevant to Riss' problem, but here Z141 can be easily applied directly w.o. error — ~~see~~ using only 1 definition per item used!

Say we have t. nym α at t. end of a binary corpus. We want to know ~~the~~ val prosty's of various reductions. We scan t. corpus for t. prefix "α" — Each time it occurred, we have various possl. contns of it in t. corpus, & each of them ~~can~~ can be t. basis of a defn. since we ~~have~~ in t. augmented corpus it will occur at least 2 times (at least 1 time in corpus, + 1 time in t. hypothetical continuation of t. corpus.

[α, of course can be of arby length. We cut off considering α of length $> l$, if ~~no~~ prefix α of length $> l$ has never occurred before in t. corpus.

Anyway, if β (= ~~some~~ contraposed α) occurs n times in t. corpus, how much do we save by defining it?

Well if β is of length l, then it is of pc maybe $\frac{1}{2}\log^* l$ for defining it.

We now have a 3 symbol code for t. corpus: 0, 1, β.

using 0, 1, β for t. precorpus; T. corpus (proper) contains: ϕ : n_0 times : 1, n_1 times ; β, n_β times.

.31 T. ~~is~~ pc of all of symbols is $\frac{n_0!\, n_1!\, n_\beta!}{(n_0+n_1+n_\beta+3-1)!}$ ← length of precorpus

.31 (R) Should we consider that corpus must end in β? This effectively ↓ n_β by 1, since n_β is ≥ 2; t. new n_β is ≥ 1.

ϕ 1 β : 1 0 0 ← corpus proper.

$\frac{1}{3}$ $\frac{1}{4}$ $\frac{2}{5}$

pc.

If we do .31 R. ; it becomes $\frac{n_0!\, n_1!\, (n_\beta-1)!}{(n_0+n_1+n_\beta+3-2)!}$

This is a rather complex way to do this simple contextual thing: But it it really gives good results ... !

The complexity arises from t. very large no. of codes for t. corpus

& its possb continuns, considering all possl. suffixes of t. corpus.
Fortunately, there's a cutoff for both future continns & for long/short suffixes considered.

If we are interested only in t. rel probs of continn 0 v.s. 1, we may be able to pool some of the continns. into 2 groups. (Those starting w. 0, those starting w. 1 (& also those starting w. β.) → which start w. 0 or 1 (one knows which)).

.05 → 19-2.21

.06 : 37.10 : So, consider this simple TM_1 : It can't learn works only by making up new defns. & "short" preds. How good could it get? I would be TM_2. After I get enuf of experience as TM_2, I will be in a better position to decide on TM_2's form, & how best to implement a completely autonomous system. { This is mindful of a self-reproducing factory that has (or too) no human in it. — Not really self-reproducing, but very useful — also useful as a proto-design for a completely self-repro. factory } → 35

.17

.18 SN On t. problem of how much (cc time) TM_2 should spend in predicting & what future problems will be. Maybe a non-cl. way: Say TM_2 has cc = C_0 (say C_0 seconds) to work on its "General Problem". & Then TM_2's task is an OZ problem.

.21 : To configure TM_1 so that it's "Expected" ΔG in t. next T seconds (T is a/ necessary user input to TM_1.) is max. This "Expected" value includes probs of future inputs. In our "C_0 seconds", we try to find codes for future continns of t. past problems, as well as modifns of TM_1 → our codes for their interactions w. thy G output

.25 are short. Hvr, it's not yet clear to me that .21–.25 is a proper OZ problem!

└ It's like t. RLP defn of probs is to try to find [a set of corpus codes w. as large a value of $\sum 2^{-l_i}$] as possl. in C_0 seconds. In .21–.25 we want to find a 'mod' (= Modifn of TM_1) → we have a set of probable future problems '[FP_i]

So that sum$_i$ (expected G of TM_1 augmented w. mod operating on FP_i · prob of FP_i) is max.

.33

.18–.33 is a possl. Breakthru; But it certainly "needs work"

.35 .17 : O.k. for Env. problems! But what about feeding TM OZ problems? how is t. P.d. involved? — How do we use 'compositions of concepts'? I may have been planning to use Lsrch for OZ problems: but say I don't at first.? I could use ~ SGA. — It may be that I can

Def

3/18/97 TM

.01 [redacted] make ti grammars used in SGA closely assoc. w. ti P.D. used for OZ probs — at ti present time, it seems like SGA's grammar would tend to be a rather narrow type ⌐design so that hyp elements could easily be generated. I don't know if my problistic functional Grammars are useable in SGA.

Another possy. is to solve OZ probs in the INV form (Apparently very non-optimal).

In SGA's grammars, the cons. in the problem can be important!

Look at some real solns to real OZ problems! — it seems likely that defns of new Concs would be important!

.09 A Broad class of soln techneques for OZ probs: first: approximate G(X) by a functional form ⌐f(x) ∋ its easy to find ≈ peak of f(x). If f(x) is "simpler" (short descns) it is more likely to be correct. This is easiest to think about in t. Continuous case, but in t. discrete case: "functional Apprxn" becomes meaningful, if we use conditional complexity as a "closeness" function (≈ a "Metric" or "distance" function).

.12 A big Q: To what extent can the "defns" of functions, concs useful for INV problems be useful for OZ probs? Are they completely difrnt worlds? ---- Seems that they should be in many ways.

.20 In Saarb, (& earlier), I made a big list of optzn. methods, w. t. idea of finding a good lang. to dcrb. all of them & then be (intra / extra) polate to new cand. oz methods.

Try to find this list! I had about 12 Methods. I found this list & have it now: include as insert: 139⅓ 139⅔

Perhaps: 1) SGA →

2) fit a quad form thru t. data & take t. peak.

3) Feed forward Neural Nets (ANN).

4) GPS (t. idea of a "metric" that is a vector) — This isn't exactly a OZ prob. Solver — its more for INV probs. —

5) Ordinary Genetic Algms. {How is SGA related to GA? Perhaps SGA was discovered when analysing GA (?).

I can maybe see how 1, 2, 5, 6, 7 are similar! See 142.05 ff for discn. → 145.02 →

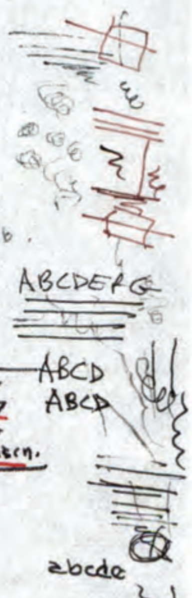
.30 6) Simulated Annealing 7) Z 141?

.31 Q: An impt. kind of OZ problem is finding short codes for a corpus!

(1) How does one do this (2) Can it be genzd. to other types of OZ probs?

Well! First one looks at t. corps! Does it have any obvious or apparent regys? — If so, test them. At first glance, one categorizes t. corpus — w. varying °'s of narrowness: eg. This is a 2 dim scene. This is a soundwave sequence. This is text from a Book.

Depending on ti source of ti corpus (& usually its associated labeling) one will have ideas about regys in t. data. A 2 dim scene can have a 2 dim fourier xfm. If it has colors, each color may have its own 2 dim spectrum. The scene may have recognizable objects in it.



6.5.97
~~7.3.91~~
Paul

Dup renamed 159 1/3 dup ~~472~~

.01:470.40

Another way to categorize P.S. (problem-solving) methods: are they for OZ, INV or Both.

1) Lsrch Both
2) AM ?
3) Case Based .... both — CB maybe related to (6) (planner) — maybe related to (6) [Planner]
4) GPS INV : Converts INV probs to a kind of "vector" H.c. prob.
5) Genz. Net of funct. srch. Both
6) Planner (.10) Both (?) ["Planner" seems to convert prob. into an "or" net → (3) case based might be looked at as a kind of "OR" approach.
7) INV (except when used to Genz Lsrch(?)) — Genzd "OR" net of tasks
8) Task net ≈ Genz. of AM
9) Gen Alg (OZ) : (But is usable for INV. w/ tricks") In this it is ~ to GPS
10) Sim anneal'g usually OZ
11) Genzd z(4) ≈ 5) ∴ Both?
≈ 9 if OSL is used

.20 12) T. H.c. method of 470.35 -.40 : ~ to (10) sim. anneal?

One approach to unifying these methods: Unify them in pairs (or triplets ...). Then Unify t. resultant genzd methods in pairs (or triplets ...)

As is, many of t. methods have already been unified

So Groups thus far,

.25 α [(1) L srch (5) Genzd network of functs (7) Genzd "OR" net of tasks
(11) Generalized z(4) (including OSL) (9) Genetic Algms (related to Sim. Anneal (10))]

β [(2) AM (8) Task net genzn. of AM .... (probly close to Lsrch ∴ α

.28 (6) Planner: Converts a task to a "OR" net. ∴ related to α
also, Related to AM, which has to recalculate pc's of next task, after it has completed work on a task

.31 → (10) Sim anneal & (9) Genetic Algms & (12) H.c. method of 470.35-.40 may be close to Sim. Anneal'g.
∴ be very Good!

(4) GPS ← so far, seems isolated.

.33 → (3) Case based P.s. .... Maybe very closely related to "Planner" & ∴ to α
See 490.15

GPS : Genzn. of a somewhat common method of solving Inv. probs. by converting them to OZ probs. The use of a vector Gorc in GPS, is, of course, a signif.

6-5-97

7-24-95 TM Genl : (OZ) : Lsearh.

Dup removed 139 2/3

OZ Lsrch is an Lsrch over the set of all optimization techniques. This set is probly not R.E. : Its probly a partial recursive set. { so do it over a r.e. set, of which optzn techniques are a sub set. ... (This strategy is what we do in A&P.) }

Anyway, to do it well : take the set of all good optzn methods one knows, & make a grammar → each is an ass'n for that Grammar. Ideally, we want a min (or short) descr of that set of objects.

We make this short descr. by defining common features, so it's easier to define the set.

I did make a list of optz. methods I knew of : In Serb. [redacted] ~1990.

Lash of that work. — (see next p : I made copy) also 7.3.91 p 472

Some methods to General Problem Solver : Use of vector Gorc. } Having a vector Gorc makes it a typical. | Having a vector Gorc makes it Atypical

There have probably been many books or equiv. written on optzn. Methods

- (2) Max gradient (a Greedy method).
- (3) Regular G.A.'s
- (4) Super G.A. (SGA)
- (5) Sim. Annealing.
- (6) linear least sq's re: soln. of simult. linear eqns, fit. local regression w. a well-fitting Quadratic form.
- (7) Lagrange Mults
- (8) stepwise Lsrch (INV lsrch) re G.A. [descbd in sel 95], Local hill climbing, using conditional complexity as a "distance" measure. (= stepwise measure)
- (9) Non-linear Oz method(s) in "Num. Recipes"
- (10) Various Books, Symposia on optzn techniques ←
- (11) Joe Traub et. al's work on optzn.
- (12) Evolutionary Pgms (vogel) (& poorman's G.A.)
- (13) Method descbd in "Science" avb. ("Tabu search" Sci 2-3-95 p 664)
- (14) ~~Method descbd by M. dela Maza, et al. — Maybe in "Red Fire" folder.~~

• Thesaurus local extrema : Useful for final "homing in" (Search & Track)

How to proceed : I combine various pairs (or triples) of methods, then combine re combinations, etc

(15) TRUST -- A deterministic Algm. for Global optzn. : Sc: 16 May 97 pp 1094 — 1097 (5/16/97)

Contains big Bibliog on Global Oz / some discussn, comparisons (in Tables 1, 2). Tabu srch. mentioned.

Ref. to a Journal on "Optimization Theory Appli", A Handbook on Global Oz methods "J. Global optzn"

5/18/97 TM LHL

If it's an audio sig: What is its spectrum, Kepstrum, Is it music or voice, or singing?
If it is speech, try to put it into ascii.
If it is text ascii text: we ask various Q's about what its apparent source & function is.

So, in coding a corpus, we first try to categorize it, using various clues: source, size,
.05 apparent symbol structure, etc. For each category that we've devised, we have special kinds
.06 of regys that we expect in corpi of that class.

We have a set of obs to hierarchically categorize corpi.

Nsa Jun 3 : "Enlarging A. Fulkerson Toolkit" by Larry Scott B.C.

.08 50 "learning" consists of (1) Getting new categories, Improving & also discarding old ones.
(2) Within each category: Obs. to make that correlate hy/g w. certain classes of
.10 regys. (Maybe an ob, of algebra?).

.11 [So: How (if at all) does A. forge. tie in w. TM devising newly defined concs., funcs, ect. based on previously defined cons, funcs?]

It is likely that A. Learning of .08 - .10 uses A. techniques of .11 — for both category construction & for looking for patterns within the categories.

A "final" category: when T. corpus fits in no known category a/o f. categories it fits best, yield no patterns. T. corpus is [i.e. apparently "Random" — So try various randomness tests — a/o use .L srch for short codes using some apparently relevant language.

.20 While 39.31 — 40.20 is certainly not an optimum way to find short codes, it is not bad, & is perhaps close to the way I find short codes.

.23 39.31 - 40.20 can be somewhat Genzed. in the following way: T. categorizer is an OB.
.25 Combinatorially On basis of the initial obs., make new obs. loop to :24 we get out of loop when certain obs are obtained.]

Also, the techniques of 39.31 → 40.20 is neither "square", "academic" — often a corpus can be in several categories. Some of the data per obs in one category may say to go to a special. other category (why pc.)

.30 Actually 39.31 — 40.20 can be easily generalized: .23 is a beginning, but for more general OZ probs than RLP, one uses similar analysis — i.e. One looks at the data of the OZ problem: This suggests obs to make to further categorize it. This categorizer → ob → categ → ob continues until one decides what optzn techniques to use. (similar to 39.31 — 4.20). Then we have the optzn techniques: We have several of them. We devise a lingo to express all of them compactly, to facilitate (inter extra) polation. We would probly do this lingo at the inter extr polation in the final compression techniques of (.05 — .06) [i.e. "special kinds of regys" for each category]

space space 142.01

5.19.97 TM LHL on SMA prob., etc. TM from 87.01 (3.31.97) 88 89

4T

.01 I'm really not so sure I solved the SMA & assoc. probs! See Nickenotes 4/28/97 item #9

item #9 (a) Given a T.S. (stochastic) & an "action alg'm" & a Gorc. —

(a1) How many params to use in the action Alg'm.?, (a2) What is expected meas[ure] of future Gorc of that Alg'm? If we have 100 data pts & 100 params in the Act. Alg'm., clearly the we are probly overfitting & our future expected Gorc is probly poor. —

But I'm not sure I had an analysis of this problem!

Look at / my previous work on this

One possl. way to deal w. .01: First: linearize the Gorc. Use the "Maxm" analysis of TM 88.23 (4/1/97 ~~4/18/97~~): ($\to 6^2 \frac{n+m}{n-m}$) ← (I'm not sure that this is really relevant)

12 Another "Complaint" about my "soln" of the SMA problem: {"soln" is: 87.18-88.10

(Jk @ work 397 6926

Consider a bunch of SMA's, w. data strings on their yields: Some of the data strings stop precipitously, giving biased estimates. In 87.18-8.10, I assumed a "probability of dropout" to deal w. this. A possl. difty: Say our data is $Y_{t-2}, Y_{t-1}, S\&P_t$ (Yield for SMA in 2 previous yrs. & this year's S&P yield) — or just Y_{t-1} to make argt. simple. The rel. of yield for t. yr. dropped out is not known. The yields can be systematically very low — which would be a reason for dropout. — or it could be not so low — that dropout could be often for reasons unrelated to yield.

20 It may be that we will ultimately be betting on SMAs w. hy & low yields in recent yrs: That selections of this kind do not drop out. (If they did drop out often historically, this would be evidence that the system will not work — ie. past good M, δ do not give "significantly better" results for next yr.!

It would be good if I could get data on yields after a SMA "dropped out" of data-list.

92.40

5.29.97 More on "SMA" problem: The present problem in Recent practical SM: —

.31 SM 81.01 & the work leading to it! TM 83.20 — 84.15: We have a joint rel. for

.32 k diff. strats: Perhaps think of this as a joint rel. of the M's of the strats. We pick the strat w. apparently large M. What is the unbiased estimate of the M of this selection? If we do the "picking" many times, we will on the average have a certain mean difference betw. the

.36 peak & the actual M of the strat selected. What is the value of this mean? What is its Varc?

I really don't have a clear understanding of this problem! I've worked out a simple case: (k indep. strats, all of M=0, & same δ). I need to work more complex cases. Also Maxm solves one kind of continuous case. What about mixes of continuous & discrete cases? What about Cover's universal portf.? — Can the Maxm method of 88.23 pp (see 92.01) use the

→ A "simplified" case for k strats: the joint dist is a Multivariate Dist. → (Spec 147.01)

5/19/97 TM

141½

.01:140.40 So; in summary; we use the usual "new conc. ands = comparisons of old ones" technique in Both the categorizn phase! (To make new obs, op.) & in the "pattern discovery" or "final optzn." phases.

Superficially, this seems close to looking for (new cond. useful things "near" old useful things, using the conditional complexity as a metric (norm).

4:56:45

4sg:

2pM

We'd like 2 features of a OZ method: ① That if a good OZ method exists, it should eventually find it (?) by studying discovery of old OZ methods; ② That it should be able to learn heurs t. way humans do & apply them to new OZ probs

I had been wondering about just what <t. actual optimality of Lsrch for INV probs> meant (assuming "all info is in P.D.")

.11 To try to apply this optimality idea to OZ Lsrch! It means that any heurs I write here could be inserted into t. P.D. for OZ srch. — just as was true for INV srch. — So t. OZ Lsrch would be truely optimal just as th. INV searchers — & for the same reasons!

TRUE & IMPT!

The mechanics of all ff are not so clear, hvr. We have to be sure that TM is actively looking for heurs for OZ probs as well as for INV probs. So this looks like a kind of "If there we assume its true, then we can prove it" situation! i.e. if we assume that TM can find any particular heur, then we can show it can solve any problem — e.g. find any particular heur.!

Hvr., not exactly: When we say "t. necy info is in t. P.D.", we assume that TM does already found t. needed heurs. T. only Q is: Can all needed heurs be put into t. P.D.?

What about "Quick Abort" heurs? How do we put them into t. P.D.?

Well, the P.D. is over all of TM's actions. Those actions could be oriented toward "Quick abort".

G.A (.05) genetic algorithms Z141 (.21)

.01 : In general, w. OZ probs., simulation of ≈ the way I seem to solve problems; 139.31—140.20 is one descrptn. The stuff following 140.20 is also relevant. The main idea is to express my own techniques in a very general way, though TM to do ≈ these things

.05 On GA.. : I had been thinking of looking for common subnets in the nets that represented successful cands. I would then define these subnets & assign them pc's. They would then tend to be used as "units" in new cands.

A possl. Trouble w. this: That once we got a subnet defined, it would occur unusually frequently in all cands. This would reinforce its occurrence in successful cands, but perhaps spuriously. It would seem like self-reinforcement/ "self confirmed hypoth"

General Thots about TM, Learning: TM, using Lsubs & its current "P.D.", solves a new problem. The prob., its soln, & soln method, are then put into a corpus, & a new P.D. is worked on for this augmented corpus. This new P.D. is then used on next problem.

.21 : 138.05 Another point on this: This "selecting subnets of the universe by frequency" is ≈ the older Z141 problem. As I remember, this didn't work quite (initially), because, after first defn., the seq. was no longer a Bern. seq.

The T did fix it (I Prof) but the resultant system was quite complex. (I never really Tested it)

Some newer thots: (1) In Z141 I only used binary defns. One could use n-tups/ w all possl.n, w all possl.n's consider all possl. parsings as 37.11—38.05.

Another (quick & dirty) soln to Z141 : When a binary defn. is made, allow it ∋ the resultant seq. is a Bern seq. (i.e. the pc's are computed assuming the new set of symbols are all indep.). This (last) is a sequence of "greedy" decisions (i.e. not so good). It could be made less greedy by considering various (sequentially decided on) pairs of binary defns (or triples of binary defns.)

38.50 .33
22.23 -.07
21.24 -.02
42.19 -.21

Note that defining new concs (items, subtrees, etc) are just part of the more general method of extrapolation by "passing a language thru the data". — In the cases I've considered, it was extrapoln. of unordered sets. — As such, the new genus of SOLOMONOFF applies, as well as "the new way of looking at it".

.39 Hmr. using "common subtrees" in G.A. is clearly an example of using a INV soln. method for an OZ problem — so G.A., in this sense, is "non-optimal".

→144.01

Why store > 1 code for Corpus?

Other than getting more exact pc's: The best code will be for an "unconditional" pc: Essentially, the average over various conditions. Any other code that is sub optimal over all of the conditions possible, is a vice candidate when one knows more conditions

In general, even conditional pc's are averages over more specific conditions — So even when estimating conditional pc's one should retain as many of the codes as is feasible.

.01: 142.40 Spac: I think this was what Schmidhu was complaining about in G.A.

So 142.39's remark about G.A. is Generalizable! What GA does At best, is pass a stock lang. thru. thy G" data. That it often works, is not surprizing, but in general, it would seem that things like SGA would be far better.

A somewhat More General form of SGA: Take all of the previous [Xi, Gi] pairs: pass a language thru it: including the Gi values as one extra component of the Xi "vector" (Xi is usually a string, but it can be a vector whose components are strings a/o real nos, a/o integers . . . .) Anyway, we have this stock Grammar & we want to find new kinds of hyps. expected Gi values.

Another, related, way: express Gi as an optimum fit function of Xi. (This is close to SGA) Then find X's w. hy extrapolated G. If, this latter operation is usually easy to do, then we have ≤ GA.

If the latter operation is not so easy, then, if finding G of a cand. X is very expensive, then it will pay us to find the peak of our approximate function.

One big diffy in the lang., is that for arby Xi vectors string/real nos. components, fitting good G approxns, is not easy.

One way (other than optimal SGA w. prob → G) is, for each G range (G ± Δ) (we divide G into "bins") we devise a stock Grammar. — Then we see how the various grammar components vary w. G. This may suggest how to do a more general G(X).

Or, make the G ranges be ± Δ, but consider Grammars for each G with a grain size << Δ — so the languages overlap, thus in one sense we "ssz".

Another possl. approach: Partition the X space in a "useful way", so that for each partition G(X) is easy to approximate (Hopefully in a way so that we can easily find X's of hy G).

While its clear how G.A. & SGA fit into the lang. driven. ——
Its not clear how it fits into (a) The Most General OZ schemes & things like OZ Lsrch (b) 139.31 – 140.40; 142.01 – . . . (The way I think I solve OZ & Inv. probs).

.35 Re: INV Lsrch: Here we pass a stock language thru all previous Inv (problem, soln) pairs (soln: matrices) and we pick soln methods (as a fn of problem) in pc/cc order.

Woops! Maybe include Time to soln. better? Lsrch does consider it — but doesn't do it the best way?

.37 OZ Lsrch: Here we pass a stock lang thru all (problem, optim method) pairs of the past. Each has had its own associated G as a function of T (≡ cc). Unclear as to whether .37 is close to what we want; what we want is a p.d. that will

assign hy// (cost pow) pc to t. optzn method that is really t. best for the present problem.

.02-139.30 SN Th. idea was to show how a large bunch of optz methods can be regarded as "" so we can "pass on long. thru them"

· The items 1, 2, 5, 6, 7 are all "similar" thru t. disn of 142.05 ff ([I'm not immediately sure about Sim. anneal, but I think it's ~ to G.A., but simpler"). [see 139⅓, 139⅔ for more OZ methods]

~~Also~~ #3, Neural nets feed forward, is a rather simple hill climbing tech reqd for fitting non-linear functions to data. It can be used for optzn, hvr. As is, it is used to optimize fit of output to designated set, but probly it can be adapted to most any continuous (can some discontinuous) optzn. problems.

#4: GPS: ~~does~~ (t. vector Gen) does seem different.

Bjarne Stroustrup

→ well look at both Inv (144.35) as well as OZ (B144.37) : Neighbors on very firm ground!

In discn. of prob. solving! Try to put all methods into a common framework. All Inv. probs can be expressed as OZ probs! (using a Hamming metric / say Pre closeness to Soln.). Many Solving $x^2 = 2$ can be usefully regarded as an OZ problem.

GPS, with its "Vector Gen" is of special interest! In digital "closeness criteria", one can have several types of "errors" each w. its own method of correction — & each w. its own method of pc assignment to t. error. A common way is Hamming metric: how many bits in error? — This gives a simple lower bnd on pc of correction.

While t. GPS Gen can always be put into t. form of a scalar (1 dim) Gen, it would seem that this misses t. point! T. final Gen is, I guess usually if not always a scalar — but t. Vector Gen is certainly more useful in prob. solving · --- in telling one "where one is" w.r.t. t. Goal!

In prob. solving: Probly best to return t. Gen as vector, as long as poss!. Combining into a single scalar [Dis Cards INFO] !

Hvr. note that t. Vector Gen is a (usually very useful) elementalization of t. problem (divide & conquer!)

Also, division of problems into OZ & ENV classes can be very artificial — The very often classifying a problem as OZ or ENV. points out its similarity to other probs in t. class & suggests Soln. methods.

Criticism of Finite Horizon analysis.

.01 130.22 One dirty I had was @ deciding how much time to spend [redacted] getting good problems on

.02 what to expect in the way of problems in the future.

Also, (minor Q): What horizon to use

It would seem that if accuracy of predn of future probs ↓ rapidly w. horizon (say $\frac{1}{2}$ like $\frac{1}{\uparrow}$) then using horizon of $\uparrow$ may be about best. Using larger horizon may [redacted] k vanishly larger than 1 (see 129.24 ff for defn of k).

* for it to be worth while to do this s.i. (self imp.) we need $k > \frac{\uparrow}{\uparrow - \uparrow_0}$

See 129.24-.40 for defns. ← 129.90

Another point: That we assume problems of f. future are pretty much like those of p. past, so f. prob. of .01-.02 is not impt.

Another way to solve .01-.02 is the "problem pool" approach to T.M.

SPEC

.01 ! 141.40 : 141,32–.36 Seems like an adequate definition of t. problem. [redacted]

92.40

We would like to end up w. a mean and σ^2 of that mean, for t. ~~strategy~~ strategy we chose.

I think t. σ^2 of that mean & the σ^2 of t. strategy itself are quite difrnt objects, but closely related.

So we estimate the mean of our strat as corrected by our "picking" operation. The varc. of this mean is also increased. The $(error)^2$ using this new mean has 2 components (as in **MeXm : 88.23 ff**) ① Due to error in t. strat. itself. ② due to uncertainty of t. true mean of t. strat. We may be able to treat these errors as in 88.23–.40 ff. } Also: see discn. of 169.18–.28

.11 To get a better understanding of t. problem: Instead of starting w. t. d.f. of t. M's of t. strats; Consider t. data that gave us that (purported) d.f. —— Then use ALP to get t. d.f.s of whatever we really want. Use t. previous discussns to suggest various exact & approxn. methods to get t. desired d.f.

.14 Start out w. case in which one has [redacted] only 2 strategies, since this is easiest to understand.

7/12/97 : A possl. approach & move toward Understanding! Δ ABCDE.

Do a Monte Carlo simulation: Possibly using Bootstrap technique.

.20 E.G. Say one has k difrnt SM strategies: How much **SOY** is obtnd by picking t. apparently "Best one"?

Say one has n months of data: Each month one has a k vector, giving t. k values of deviations from t. mean of t. k strats. } Is this what I really want ??

.24 One way to do Bootstrap: From t. n [redacted] months: Select random sub-sets of m months. (Clearly (as one can see by making m close to n) there is some serious statistical error here. Even if k << n, so trials are more or less indip", one ends up w. t.

.26

.27 result for m months, rather than t. desired, n months.

.28 But anyway: t. Q is: [redacted] : w. this m subset selection: what is expected μ & σ

29 of peak yield of t. k strats, minus t. mean yield of t. ~~[illegible]~~ strat. w. that peak?

[If we understood this problem better, in .24, we could "correct" for m not being << n; & also, we could extrapolate the results for m to t. desired result for n

I think t. Good Part is .20–.29: This gives (in particular .28–.29) a clear picture of just what the problem is.

.35 On t. nth root of a D.F. = Defn: if [illegible] A = nth convolution of B then A is t "nth power of B". So we take $\mathcal{F}$ of A, take nth root. Then ~~[illegible]~~ $\mathcal{F}^{-1}$: We can get any power we like, this way. For a k variables, taking FFT & FFT^{-1} does take a lot of time, but is doable.

Is there good way to take ~~[illegible]~~ FFT of a no. of pts ≠ $2^{N(\text{integer})}$?

(162.01 spec) →

(153.01)

Three immediate Problem Areas!

1) **SMA** problem: Its a fairly general problem of evaluating evidence (147.01 - .14)
(≡ PATH problem)

2) General plan of TM: Is t. schema of Sol 86, 89 viable? To what extent can I say that I have a general ~~way~~ method to solve almost all INV, OZ probs? Initial Immediate Motivation was a possl. lrng. discn w "PUSH" (Marvin's Grstudent)
Latest Part! 133.01 — 140.40: 141½.01 - .40, 142.01 - .40, 144.01 - 145.40.
This is a continn. of an earlier seq. starting ~ 1.01

3) Analysis of Schmhv: I want to write a good analysis/review of his papers — to him & to myself. 101.01 — 131.40

I Guess what I want is a quick, easily Readable summary of what t. problem(s) is (are) ■ T. most promising lines of attack: Most recent bottlenecks.

.10 Some Quick notes!

on ② : ⓐ Can I take any desirable problem soln. (technique included) & express it as a special case of Lsrch? ~~[illegible]~~ (ⓑ is a subproblem of this part.

ⓑ Can I express all heurs I can think of as learnable modifns of t. "P.D."?

[Here I mean "P.D." in t. more general sense: including ~~[illegible]~~ cc. as well as pc] → see 136.03

ⓒ If ⓐ & ⓑ are "under control" ~~[illegible]~~ can I organize much of t. / existing work on "Expert Systems" (in learnable form) & Machine lrng. into a tng envt. for an eventually very smart TM.? [↑ (I guess what I mean by this is that I regard a particular E.S. as a mode of behavior that T.M. has to learn, from a suitable t seq)

→ Also include other tng envts: like text books (perhaps modifyd for TM's training) or any other tng. envts usd for humans or Machines

---

.24 "PUSH" suggests / set of Benchmark "Mach. Lrng" problems on t. NET:

It may be ~~that~~ .10 - .24 is pretty much my present approach ~~&~~ to TM. — That

.30 133.01 ff is concerned w. "details" ..... but I'm not sure: 133.01 ff does seem to have lots of ideas on ~~doing~~ working various of t. sub. problems of ②ⓐ, ⓑ, & ⓒ

.10 - .24 seems to make it all very clear & "simple": but 133.01 did seem to uncover some very serious problems: What were they?

Some Cands: ① 136.24 ② How to implement "Quick Abort" as a PD modifn.

③ What is t. P.D. ② P.D. of? ④ How can we merge t. P.D. for Lsrch w. t. Pd for OZ if t. P.D. for OZ is part of a SGA system & is ∴ a P.D. for an entirely difnt kind of Univer? → 138.17

⑤ 136.38 When is one PD "Better" than another? (relevant to TM2 trying to improve TM1's "P.D.") ⑥ 137.01 : common subtrees in solns. were used in hyper pc in t. P.D. — but this didn't implement "1 Shot lrng" (like Case Based Lrng).
(40.11 seems more relevant)

.36 ⑦ Z141 ~~as~~ as / originally written didn't work. ~~[illegible]~~ — This was a critical idea in defining new concs

140.21
137.11 ff has some ideas on how to (improve/make workable) Z141 — one way is to use lrng means.

→ But this doesn't face t. problem of ~~[illegible]~~ / cand concs being compositions of ~~[illegible]~~ useful old concs.

⑧ 139.31 goes into t. impt. common OZ problem of finding short codes for a corpus.

6.6.97 TM Gen. Plan. Z141

.01 149.36 : on Z141 Seems impt. I do need a way to define new concs as compositions

.02 of old concs. I think t. main trouble w. a simple soln. to this was that if an ngm had unusually low frequency, we had no way to take advantage of this .... it was a clear "regularity", yet we weren't able to put it into t. grammar — so it was, essentially "invisible" to us.

Thinking about it, hrr., I'm not sure ~~[illegible]~~ Purpp. remarks was correct. If a certain ngram occurs in unusually low frequency, Then maybe one could by corpus cost by defining it. ! — a skill keep t. seq. of symbols Bernoulli!

.09 A few ways in which this mite occure: AB occurs very infrequly (maybe never never)

If $\alpha \equiv CA$, $\beta \equiv BA$, $\gamma = AB$ have been defined, & A, B, & C have very low freqs in t. corpus : e.g. it's mainly α & β: Then we have ↑ pc of corpus w. this defn. Can we express t. unusually low freq of AB in t. form of a Bern Seq. of suitable symbols?

A finite state system mite be easier to use to model this reply; but still need — I want .01-.02 ! Well .01-.02 doesn't actually need "ngms of unusually low freq".

Defining ngms (or functs. or operators) has at least 2 functions: (1) shorter code for corpus : good predictive (2) Use of resultant Grammar to generate new trials of reasonably by expected Utility.

20 So we want a grammar type in which we can use t. info that if "A occurs, B is unlikely to follow" (or, if B occurs, A probably to precede it) Clearly, a finite State Grammar can do that. When "A" occurs, we get into this particular state, in which t. probly that "B" will follow is unusually low.

Can I have a finite state Grammar & also define "concs" "objects"? O.K.! In a FSG, we have defns. of states & defns of t. things t. states induce.

Or: perhaps just look at one's normal way of defining concs & composing them. Perhaps my ideas about composition are based on "unordered object" inventions, rather than / in finite sequences extrapoln.

Defining a (very low w) "AB" can ↑ properly if Reader is no other way to get "AB" other than thru t. defined symbol: e.g. while "A" may be a possl. (macro) symbol, "B" should not be.

.35 In many statistical studies; (Industrial), we want to know whether t. probly of a certain event class is $< 10^{-6}$ or 10^{-10}, say (Nuc meltdown, Bridge collapses ...). T. way we usually do this is we make a model in which t. event can only occure in certain ways. We then show t. EPC of all of t. ways we can have time to consider, is < threshold.

Perhaps in the spirit of 149.33 (a bit like 149.29), we may have to arrange a grammar so that the only way we can get a certain kind of event is thru a certain set of codes — & those codes are of very low pc.

So, the fSG of 149.20 is one way to get low ω for AB.

Another way: Make any other kind of Grammar in which all of the ways we can get AB are known & of Σ, low pc. E.g. Born seq. in which $\alpha \equiv AB$ is the only way to get the seq. AB & α is of low pc.

This will be complex, however! Say the α part is A, B, C, D: we start w. corpus: A seq. of those symbols. We first note the freqs of A, B, C, D, & write a compressed form, using those freqs. We next note that $f_{AB} \ll f_A \cdot f_B$.

How to proceed? To make a set of forms so $\alpha = AB$ is the only way to get AB.

We could do this by defining AB, BB, CB, DB, — first looking for grammar universal (?)

We now have only those symbols: unfortunately, the thing is now incomplete...

No way to express BBC! Well, yes! We parse the corpus 2 possible ways:

$(X_1 X_2 | X_3 X_4 | X_5 X_6)$ or $\cdots X_1 | X_2 X_3 | X_4 X_5 | X_6 \cdots$ Then will still

We then need 6×6 = 36 dual symbols! Not so good!

A denumerative idea may be 149.35 — that we have to restrict the freq. so only one or a few ways to say "AB".

It may well be that Born seq. can't do what I want it to do!

Even tho' if AB has unusually high freq, defining α = AB, but we still have the possy of parsing AB as AB rather than α(≡AB). So if we define α ≡ AB; the α's empirical freq will be the old $f_{AB} = f_A \cdot f_B$, because we still have the parallel parse/enumeration of "AB" as "α" or as "A, B"

— Maybe not! The codes in which all "AB"'s are coded as "α" will be shown much more wf. than the codes using "AB" → "A, B".

.29 Suppose we dropped the "low AB" freq. problem for a while & just defined by ω "AB"'s. Well we could end up w. operators in which we should note that the second argt. is merely an 'integer' or never a string.

— But more commonly we notice that the second argt. is almost always a string.

.33 .29 suggests: Born seqs have uncond. probbys for symbols: we could modify this (if we have one of 55Z) to do cond'l. probys. "Low freq of AB" could be easily implemented that way! — Also argts. of a function could be conditional on the function (as opposed to being uncond'l. probys).

Well! One way to do "low ω AB": we have already defined uncond'l. probys for various symbols — some of which may be "ngms". Hvr. when A occurs, we can make a special situation of conditionality: B is given a lower prob'ty & the rest of the symbol probys are kept in same ratio, but renormed.

6·8·97 TM Gen Plan: Z141.

Pg 149 - 150 lost 6-9-97.

.01 150.33ff is a bit asymmetrical! For hy w we use dscrtg of npms; for low w we use condl probys! It would seem that condl proby is more powerful because it can handle both/mixture low w. I could just try either/both; we can define npms that have them wholly or partially conditionally probable.

Probly it would be well to try it in some problem envt. to get a better idea of what I need.

Maybe 11 input Mux?

12.58

SN GPS is not really an "OZ" problem. The reason is that the global peak is desired; for this reason, each component of the Gorc can have any wt., as long as the wt. is >0. The only way to find the global/min is for all components of the Gorc to be zero.

However, when we have Gorc components it can be regarded as "sort of" an "dependent" "OZ problems: But in regular OZ problems we have a fixed time to get work. It's more like an "Anytime" problem — get as good a soln as one can in available time, but keep

Working at it!

General Plan: When TM has worked any problem, it is "Bias" should change. Also Expo This should be true for whatever kind of prob-solving we do. E.g. G.A., ANN, LSrch (changed Bias after LSrch is an additional pem — [redacted] — except for OZ LSrch, perhaps)

Anyway — how to best shift this bias is one problem!

But the TSP problem involves sequencing problems (or performances) of TM so that the bias obtained from one problem is useful for solving the next.

In using LSrch to solve probs in the G-A. way, we end up with a bias in the TM's Pd. That corresponds to the bias in the G.A.'s "population of seeds".

Outline of General plan!

1) 2 kinds of probs: (a) Inv. (b) OZ : Give many examples: Also show variety of OZ probs.

2) Induction, Lrng, probly: as a kind of OZ problem ← (1)

3) L such as Soln. to both (a) & (b)

4) expressing Expert systems that have been constructed, as "learning" problems: What corpus, trg. seq, trng envt., probs, hints, reading, hours, would be needed to discover ("learn") & desired behavior.

5) T. General problem of educating a machine that has an "Adequate" lrng Alg. (case based lrng.)

(a) Can we use much of existing A_I, work in E.S., Machine lrng, Patt. recogn., Discovery Systems, G.A., ANN, etc. — for training & same "instant TM" (yes!)

6) Use of Math universe of proper types avoids 2 (inessential) diffys: (a) Optical, acoustic into proc: vonx computationally intensive

(b) Much of "commonsense" background needed for Physical R.W. & language lrng.

7) How to teach it Language: First it learns a part of math. Then a simple Eng. Eng, w. discussions about what it already knows. Then teach it new parts of math & physical sciences perhaps & expand E. language as it has learned.

8) Lang. lrng. as very impt. goal: It can make teaching much easier.

9) Very fast machine, means we can put larger conceptual jump steps between in trg. seq.

10) The "working backward" approach to Trg. Envt. design.

11) Koza's methods in G.A. After solving a problem, wipe out all learning & start on a new problem (Academic pressure to prove "learning").

12) Criterion of progress: Able to learn hard probs, easy to learn, not Brittle! Can solve probs it was not designed for.

(13) Look at 148.30 ff (from 133.01 ff) planning

(14) Try to keep the top general Goals/patterns of lrng. in mind at all times! Keep it "In English" for a long time!

6.17.97 TM ~ SOY Spurious Optzn. Yield 153

.01: 197.40 In [redacted] MQ I got 7000 difrnt Models: & Test set had mean err² 4× that of test set! 1

$N \approx 450$, $k \sim 60$. $\frac{N+k}{N-k} \sim 1.31$. only!

If this pgm is correct (it may not be!) could t. 7000 models account for [redacted] much of t. factor of 4?

In Maxm, the apparently best comb. of coefs is t. best bet, but the apparent (ms sqare)

.07 gore is too small by factor $\frac{N+K}{N-k}$. → .26

In Cover's portfolio, He does not pick t. peak, (but it may well be that what he picks is close to t. peak). Hvr, his estimate of yield may be closer to correct.

vare of $\hat{\sigma}^2$ estimate? Pick best of > k.

T. best policy proby would be to pick $\vec{b}$ w. max past yield, — but determine in some way, expected future yield, as discounted by "optimization penalty" **

Saturday Th. Fri

2 → Suppose near t. peak $\vec{b}$, we have made studies of variation of yield, (daily, Monthly, yearly). From these studies, we decide that at $\vec{b}^*$ (= peak $\vec{b}$) the yield is $\pm$ σ_y.

(Shouldn't hyer yields be less likely, so t. d.f. is not symmetric?)

(But wrong?): This transforms th. distribn of yield v.s. $\vec{b}$ into proby's peaks. We assume that near peak, t. σ_y's are about t. same..}

Once we xfm it into a proby distribn on t. params: It is ~t. same form as Maxm! — we then compute the "unbiased error" & map this into unbiased yield!

Actually, t. correspondence betwn t. "SM" prob & Maxm is closer: Maxm needn't have probys!!

The function in Maxm gives error² as a fnctn of params; "SM" gives Yield as a fnct of params.

.26:07 → For each network we get a $\overline{error^2}$ but each such estimate [redacted] has its own σ^2.

So we have 7000 difrnt error² distribns: each w. its own μ & σ; we pick t. net w. min μ (of err² d.f.). So this is t. usual OZ penalty problem (OZPP)

Def? The "7000" is not so impt.: What is really important is t. no of nets with their μ's that are within σ's of t. minimum

on SOY → 162.01

6.17.97 TM

Notes on the d.f. of the var. obtained for a normally distributed random variable:

Cramer: fr. f = frequency distrbn.

d.f. = $\int$ fr.f. = distrbn funct.

c.f. = $\int_{-\infty}^{+\infty} e^{-\gamma t z} dF(z)$ (159)

$F(z)$ = d.f. of z

c.f. = $\int_{-\infty}^{\infty} e^{-\gamma t z} f(z) dz$

$f(z)$: fr.f. of z.

p 233 If X is a random var w. ϕ mean, $\sigma^2 = 1$,

fr.f of X^2 is $\frac{1}{\sqrt{2\pi x}} e^{-\frac{x}{2}}$ ("by 15.1.4 : p 168)

Cramer takes the c.f. of the x^2, Then, takes the C.F. of x of them's multiplied them together (indiv.), Then uses c. f.'s them to get

$\frac{1}{2^{\frac{n}{2}} \Gamma\left(\frac{n}{2}\right)} x^{\frac{n}{2}-1} e^{-\frac{x}{2}}$ for the fr. f. of σ^2.

$\chi^2 \equiv \sum_{i=1}^{n} x_i^2$ (the x_i are normal $(0,1)$ vars)

$E(\chi^2) = n$, $D^2(\chi^2) = 2n$

D^2 is second moment about mean, p 180

p 234 eq. 27.4.2.

p $E(\xi)$ = mean of var

$D(\xi)$ = s.d. of var

However, these are not what I want! They are the distribn of the $\sum x_i^2$.

I think I want the d.f. for $\frac{\sum x_i}{n}$ —

not its mean, but its variance. $D^2(\chi^2)$ seems to be it, but

It should get narrower as $n \uparrow$

$x^{\frac{n}{2}-1} e^{-\frac{nx}{2}}$ / p 236 does get narrower as $n \uparrow$.

C, N

7/2/97 TMM LHL

~~153~~
155

.01 : 131.40 : A diffy m analysing J's system: He is "hill climbing" in proby distribn. space, but in a changing environment. A Just vote now, I don't remember how it worked, but

Hill climbing in a changing envt. can work if t. envt. changes "slowly enuf".
What "slowly enuf" means is rather variable. If t. couplings of t. envt. to the hill climber is weak, t. envt. can change much, but it effects on t. hic. processes will be small.

More generally, t. changes in t. envt. must be of t. sort that t. Hill climber can deal with — ie. The pgms to deal w. those changes must be of reasonable pc.

But to understand this better, I need to re-familiarize myself w. J's system.

Another approach: The hill climber tries to model t. changing envt. — tries to predict t. envt (as a kind of regression problem) — Then using its model of t. future, it has a "static" problem to solve.

7.4.97 TM: LHL

tol: N 125.01 → 90
128.20 → .25 : Re: $\vec{M}$, $\vec{W}$: $\vec{M}$ is prob'y vector defining pdf density. $\vec{W}$ is ~~working memory~~ Working Memory

So we try to find an optimum $\vec{M}$ pt., but $\vec{W}$ is changing! It may be that there is a way to deal w. this: Certain regions of $\vec{W}$ could be selected to change slowly.
— But in general; J's method of doing prob'ys, makes it hard (if not impossl) to store anything permanently!

Actually J is considering slowly changing problems (external envt.), in addition to changing f. external envt, $\vec{W}$.

J' mentioned someone having a non-double indirect memory ref. scheme: find where J talks about it: Ask about it!

78
82
252

A sort of Review of J's LHL paper:

.02 1) A criticism of J's system was w. probability 1, t. system would eventually destroy (by (over writing)) any particular good, useful part of $\vec{W}$ (t. workspace).
T. system has no good way to store info:
It might do this by writing impt. info. in many places, to ↓ proby of destruction
Another way would be to ommit J's lower limit on probys. (also his upper limit)
.09 T. techniques he uses will not give probys of 0 or 1, but can give probys arby close to 0 or 1.

2) Another serious criticism of his original LHL paper: T. probabilistic Grammar he is using is poor. Its really close to "Tabula Rosa."
What I'd like is some way for t. system to preserve routines a. subroutines that have been useful in t. past. J. is very non-committal about this, hvr. — He just retains entire pgms that have been useful in t. past.
.18 So t. Q is: If after working one kind of problem, we give t. machine a new problem, that requires soln. elements of t. soln. to t. first, & we retain t. working memory & stack of t. first problem: would it be able to learn
.21 to work t. second problem faster than w.o. t. $\vec{W}$ & stack ($\vec{P}$)?
Actually I really don't know how bad his lang. is! Its just that I think my "functional lang" is better because it/ tries defining new functs that are compositions of old functs (& functionals). — If J's lang. can do .18 – .21, then its not bad! If his lang is universal, it could eventually do .18 – .21, but t. Q is, how much training does it need before it starts doing that?
I think J. would say — we really don't know .18 – .21 is t. best way to do things — why not let t. machine figure out t. best way by itself>
— Answer: it will take too long. (N.B. → Also starting w. a random UMC. is no. v.g. Tho its universal, it could take only long. to discover a useful "Learning Strategy".)

3) If a particular SMP, SO get a very high/spuriously score by accident, then SMP's following it will not be able to meet its level of apparent excellence. So we pop back to SO; we continue using it a while — & eventually its mean $\frac{\Delta R}{\Delta T}$ ↓ to a realistic level.
So this is not a criticism! It simply shows that t. system will deal w. this problem properly.

4) AH! finally found it! P10.28 LHL "No Need for "Discount Factor" — discounting expectations of future payoff for systems w. infinite life" His "comment #8."
This seems very wrong! see 129.01 ff for my Analysis.

5) "Life is one way" Remark #14 (PP 11 & 12) may explain this. Perhaps t. idea is that events never repeat. Each event occurs one time only.
How J. may have gotten into this frame of mind: In recurrent neural nets, One has a sequence of problems in real time that t. net has to

7/8/97 TM

On problem of how much cc. TM should spend on trying to predict nature of future problems so it knows what knowledge areas to give more wt. to, i to develope.

It may well be that TM is always trying to predict future as far as it can! I.e. In ALP, we try to find codes that go into early distant future.

Hvr., actually, I may not want to spend much time on that problem; The "problem pool" approach seems pretty good & looks like it would be adequate for Training. & V.G. TM.

.01: 157.40 learn to deal w. (say human speech recogn.).

It's discouraging "Lsrch being for repeatable events", while inappropriate as a criticism, is still "correct". Hrm., does there exist an algm. anything like Lsrch, that could work w. a continuous data stream? (← Like a recurrent neural net.)

For most difficult intellectual problems (like d.t. Design of an intelligent machine), one usually cuts off external stimulus while one is working on it. External info (new work a/o criticism by colleagues) is allowed to enter the work space — but usually slowly, and at a very controlled rate. One tries many distinct candy parts same problem before modifying the problem. — eventually, the problem may become rather constant — not vary at all w. time.

To use Lsrch in a changing problem envt.: One uses a sequence of trials for a constant or unchanging problem for a certain amount of time. One obtains a soln. with a certain "appropriateness" (≡ Gorc). A new problem or modifn. of old one enters from envt. One now tries modifns of solutions to old problems. SCGDx Trials are made in order of "algorithmic distance" from the solutions to old problems. ← conditional Complexity

If new info comes in very rapidly, one will not have time to solve a problem, before its soln. becomes obsolete (no longer of interest). Rapidly changing environments of this kind cannot be dealt w. by most systems. Organic evolution / extrapolation adaptive technique can't deal w. a sufficiently rapidly changing problem sequence. "Rapidly changing" (as I define it) means that the regularities for today are so far from those of yesterday, that algorithmic distance search or descent is not helpful — (nor is anything else!)

One can actually do Lsrch on a time series as new info is coming in. Say one is looking for regularities in a T.S. An example would be SM prodn: Each instant trade gives new info. Between trades one could do many trials, but Say one wants a fit for most recent data: As a new trade comes in, one uses the best models thus far on the New data plus much of the old (almost all of the old — Say R window or k window). 109739.10 15 4000

→ 161.9

1) As we have this corpus: we write a large [redacted] FS Grammars (or maybe safer fs Grammars) for it. If the corpus is really a CFL would there be patterns in the Grammars that would be more visible than the CFG patterns in the corpus itself?

In General, what kinds of (styles, codings, representations) of the corpus would make its CFG structures more obvious?

2) In the paris paper, I managed to express a CFL as a "kind of" FSG.

As it is, the picture I presented was very much like a "hidden Markov model"

(= [redacted] FSG)

so it has ∝ pts, a multiplicity of A, A' leaves + space in between.

3) → In CT Do both Left & RT substitution forms of FSG:

If a CFG has only (Rt / Left) substitutions, it is a FSG. So we will do T. FSG formulations in both forms — Rem perhaps, combine Rems,

7/11/97 TM-LHR

Where is 1603 does it exist? If exists dated 7/11/97

161

21: 159.40 General draft of Letter to J:

1) Explain how his idea of the problems [redacted] a good non-ad hoc formulation,
This is a good formulation of a general problem of learning in R.W.

TM PSGD

PS GD
Phrase Str. Grammar Discovery

a) Non-repeated tasks

b) Prob Couples is not divide-and-conquer problems.

c) [redacted] Gen. can change in unforeseen (unforeseeable) ways.

d)

The soln. he proposes does, indeed, address these problems.

How Might this particular technique be applied, [redacted] ?

One way: to start with minimal Bias (as in his example 1) and subject it system to a very long training sequence: Much as organic evolution.

This would be very expensive, time etc-wise [via of parallel comp. the organizing of this kind for computational training.

Alternatively, start with more bias & a more carefully designed TSP.

My own interest is in a somewhat distant problem: Designing a very intelligent robot for solving a great variety of difficult problems.

To achieve this, I train/begin by the system on well defined problems. The system can make many trial solutions to a particular problem. The goodness of it solution [redacted] is in general dependant open about how much time or other resources are used it used to solve the problem, as well as how useful the solution itself is.

For many problems, L search is very good, if applied properly,

I believe that it is possible to start with a system that is only able to solve well defined problems — and is able to use past trial solutions for its problems — Then train such a system to solve problems in more realistic R.W. environments.

dynamics associates

.01': (spec 147.40 ~153.40) → Hvr., how does one take a fft of a pt. d.f.? — Actually, it mite be easier, faster, to take a regular $\mathcal{F}$: perhaps first make a table of $\sin(x)$. An easy way to get $\sin(nx)$, given $\sin x$, for many n:

$e^{ix} = \cos x + i \sin x$; $e^{inx} = e^{inx + ix} = (\cos nx + i \sin nx)(\cos x + i \sin x)$

So from $\cos nx$, $\sin nx$, we can get $\cos(n+1)x$, $\sin n(x)$.

This would give the $\mathcal{F}$ of a D.F.: Then (may be) take $\frac{1}{n}$ power, then take $\mathcal{F}R\mathcal{F}^{-1}$ of result:

Try it for $n = 1$ & see if we end up w. an approximation of original density function.

The reason for interest in 147.35 (i.e. nth power/root of a d.f.) is to make a correction for 147.27–.28: i.e. If we had a soln. for m, how to extrapolate to get soln. for n? ... The Phy is not so clear: — see .30 for idea.

Hvr. on a Theoretical front: The reason "characteristic functions" are so often used in Probability Theory is that convolution of d.f.'s is a common operation, & it translates into multiplication of corresponding char. functs. ("Char functs" are $\mathcal{F}$'s as far as I can see.)

Anyway, to find the expected value of the peak of a d.f. of 2 variables; This may be expressable as a convolution! This idea can be useful in at least 2 ways: 1) Theoretical analysis of Soy of d.f.'s — so I can know the functional form of it & extrapolate to large (say 30) k.

2) I may be able to go from m to n in 147.26–.27 in an intelligent way.

---

30 Actually, what happends: say we have a d.f. for M yrs: by $\mathcal{F}$ & $\mathcal{F}^{-1}$, we can get d.f. for n years (say d_m is d.f. for m yrs: $d_n = \left(\mathcal{F}^{-1}\left(\mathcal{F}(d_m)^{\frac{1}{m}}\right)^n\right)$

→ Some confusion in my mind betw. effect of ↑ of $m \to n$ (↑ no. of data pts) v.s. changes in k (i.e. no. of strategies)

163.2.1 →

Perhaps the usual way to code a simple scalar or vector linear regression:
To code the $N+1^{\text{st}}$ item, (One has already coded the previous N items)
from the previous N items one makes a fcst. for the next item. This can be simple
(linear regression or coding of k dim vector: predn can be linear or non-Linear)
One simply uses the pc of the next symbol: But MLE has to be an "unbiased
estimate." — so use $= \frac{N+k}{N-k}\sigma^2_{\text{post}}$ for simple scalar regression.

Another interesting case is Multilinear regression w. ≅ log. The prediction
.09 has a mean of $M_i = \frac{1}{N}\sum_{t=1}^{N} P_i(t)$: $i = 1/k$ (k dim. vector $\vec{M}$).
.10 Its "Vare" is $\frac{1}{N}\sum_{t=1}^{N}(P_i(t)-M_i)(P_j(t)-M_j) = M_{ij}$ = cov. matrix.
.11 M_{ij} has to be mult by something like $\frac{N+k}{N-k}$ to get an unbiased estimate — as
remarked.
We must be able to empirically check on the $\frac{N+k}{N-k}$ factor in the last case.

The main point I originally wanted to make was that we have no yeast for the
Covariance Matrix (or its 1 dim. equivt.).

The coding of the first few items is distant from Tforpg., but if we use
X window (& perhaps even R window), that is irrelevant. ... unnecy.

.2((on SOY) 162.40 In spirit, the Tforpg. is relevant to SOY: Here, we want a k dim.
multivariate d.f. for k strategies. Say we have daily data on yields of each,
& we want to know the SOY assoc. w. picking the apparently "best" one.
We obtain the d.f. via .09, .10, .11 (including $\frac{N+k}{N-k}$ or whatever).
From this M.V.D. we can then do a Monte Carlo to determine the mean & vare of
the SOY!

To get a Monte Carlo d.f. for a set of indep vars! (Diagonal cov. Matrix)
is trivial: To get it for a non-Diag. cov. Matrix: first diagonalize it — obtaining
a basis set of new strategies: The new strats are orth to each other & the
old strats are (known) linear combns. of them.

T. Tforpg involves diagonalizing a matrix, getting its eivecs & eivals.
I big job (unless Maple can be persuaded to write strings for it)!

An easier way that may, in one way, be better theoretically,
& certainly (easier / simpler) to implement:
Say $\{P_i(N)\}$ $(N = 1(1)\infty;\ i = 1(1)k)$ is the original data: We make a new data set:

.39 $$P_i'(N) \equiv (P_i(N) - M_i(N)) \cdot \sqrt{\frac{N+k}{(N-k)(N)}} + M_i(N)$$

$$M_i(N) = \frac{1}{N}\sum_{t=1}^{N} P_i(t)$$

See 164.18 - .28 on "$\frac{N+k}{N-1}$": Prob & Spec 164.01

spec

.01: 163.40: We then use the data set [$p_i'(t)$] in the following way: Each day/t we select the $p_i'(t)$ that is largest of the N stocks: say, this is $p_{j(t)}'(t)$:

.03 we then get $\frac{1}{N}\sum_{t=1}^{N}\left(p_{j(t)}'(t) - M_x(t)\right) = \underline{\underline{SOY}}$

we also get expected error in SOY.

.05 This is $\pm\frac{1}{N}\sum_{t=1}^{N}\left(p_{j(t)}'(t) - M_x(t) - SOY\right)^2$. — Maybe mult by $\frac{N+1}{N-1}$?

One simple way to check on these formulae:

.17 First do $M=0$, $k=1$; $M=0$, $k=2$. Mainly check $\frac{N+1}{N-1}$ & $\frac{N+k}{N-k}$ factors

953
Dav [redacted] M
GaN 162M
CE 111M
PdPS 323+
LA 619+

.18 SN Related $\frac{N+k}{N-k}$ in 163.39 & 164.17 : (? Note .18-.28!)

Say one has an actual M.V.D. w. a certain mean & certain cov. matrix; we take a sample mean & cov. assoc. w. a sample of N examples.

How should we correct our cov. matrix to make it unbiased? → if we had the true mean, C_S would be an unbiased estimate of C_T

For starters, the mean is in error; the d.f. of error^2/ is given by $\frac{1}{N}$ · the cov. matrix.

Say C_T is "True cov. matrix"; C_S is the 'sampled' cov. matrix.

$C_T = C_S + \frac{1}{N}\cdot C_T$. : $C_T\left(1-\frac{1}{N}\right) = C_S$: $C_T = \frac{C_S}{1-\frac{1}{N}} = \frac{N}{N-1}\cdot C_S$.

Then, say C_{SN} is the cov. matrix for the next element in the sequence,

is $> C_T$, because the mean is in error; so $C_{SN} = C_T + C_T\cdot\frac{1}{N} = C_T\frac{N+1}{N} = \frac{N}{N-1}C_S\cdot\frac{N+1}{N}$

$$C_{SN} = \frac{N+1}{N-1}\cdot C_S$$

.28 N.B. K does not seem to appear in this formula! ←

The formula of 163.39 can be simplified if we multiply $P_i'(N)$ by the constant factor $\frac{N(N-K)}{N+K}$: giving

Maybe use $\frac{N-K}{N+1}$ here; see (? .18-.28)

.32 $P_i''(N) = P_i(N) - M_i(N)\left(\sqrt{\frac{N(N-K)}{N+K}} - 1\right)$

.33 If, instead of $M_i(N)$, we use $M_i(t)$ the correction is constant for the entire data set. — simplifying calcn. very much.

.36 N.B. the formulae 163.39 ; 164.03,05 assume the data pts, $P_i(N)$ are uncorrelated sequentially : otherwise the cov. matrix for k observations will not be k times as large as the cov. matrix for 1 observation.

This sequencial correln. certainly is important. To t. extent it exists, we can do prediction. If the $P_i(t)$ are already predictions, using even if we use all previous data, to optimum advantage t. preds themselves we can be given correlated.

The big Q is: to what extent is t. $\sqrt{N}$ factor a measure of t. dilation of the cov. matrix for larger N? If there is correlation, does this effect the relationship of cov. matrices for N=1 v.s. N=T, in a simple proportional way?

One could plot the size of t. cov. matrix (or simply it 'trace' ≡ sum of diag. elements) as a function of N. Or, Do a window of width, T for generality, cov. matrices —

SN T. discn of correlns. at 164.36 ff. started when I considered using t. Soy analysis on Cover's "optimum" portfolios. I would use $b(t+1) = b^*(t)$: & see how bad it was. — But remember "Soy" was originally designed for max yield, assuming a leverage = 1 — so vere was not considered: I don't know if Cover really considers leverage in a realistic way.

Variation of cov. matrix w. Δt:

1) I would change notation: N is t. entire corpus length, T is a pr. within t. corpus: Please the new notation is opposite from that of 163.39 ff.

.23 2) The problem is: If $P_i(t)$ is t. yield betw $t-T$ & t, we want to know t. cov. matrix for $P_i(t)$ & see how it varys w. T. By approximately "Matrix size" as a linear function of $\sqrt{T}$ we can know how to Modify "N" in 163.39 ff: $\sqrt{N} \to \sqrt{r \cdot N}$, where r is a correction factor. $r<1$ for positive correly $r>1$ for neg. correln. (or vice versa?) → see 166.01 for t. Move on "r"

3) In addition to t. diffy of 2) (.23) There may be a different "r" for each of the k kinds of strategies! How to work this in, is quite unclear! One way: in (64.32: use

$$P_i'(t) = P_i(t) - M_i(N)\left(\sqrt{r_i N}\cdot\sqrt{\frac{N-1}{N+1}} - 1\right)$$

(say constant, like in 164.33)

Not Bad! Each $r_i N$ ($i=1|k$) tells how correlated are its values: $\sqrt{r_i N}$ is the factor that relates t. D.F. per 1 trial to t. D.F. for the sum of N trials' — it is different for each of the k strats, & can be computed for each individual strat.

7/14/97 TM SOY/SMA

.01 Actually, calculating the r_i is not so simple. We want ${}^N\sigma_i^2 = N r_i \sigma_i^2$

Here σ_i^2 is the var. of the i^{th} strategy : daily.
${}^N\sigma_i^2$ is " " " " " : for N days.

While σ_i^2 is easy to est. usually, ${}^N\sigma_i^2$ has to be estimated:

— This is the problem of 165.23 ; w. $\uparrow = N$. We can get ${}^\uparrow\sigma^2$ for each value of $\uparrow$, but SS_2 $\downarrow$ as $\uparrow$ approaches N. In fact, SS_2 is $\simeq \frac{N}{\uparrow}$.
So one could obtain ${}^\uparrow\sigma^2$ for smallish values of $\uparrow$, then (crossly) extrapolate to $\uparrow = N$. This extrapoln isn't too bad, if we found ${}^\uparrow\sigma_i^2 \propto r_i \uparrow\sigma_i^2$ — so we just to a "Best fit" for r_i. We can do a least squares fit for r_i, using wts. of SS_2. (But also, we want more wt for large $\uparrow$, since we really want ${}^N r_i$.)

So: T. main problem is: What is relation of d.f. for single day, to D.f. for several years' corpus?: I.e. the corpus on which one decides which strat. to bet on.

The means of the 2 d.f.'s are of necessity, related in a simple way. The big Q is the 2 Cov. Matrix's for 1 day v.s. several yrs.
This Q is addressed in 165.32-ff, for the "Jackknife" Monte-Carlo Simulation, but the problem occurs in the same way, if we attempt to do it "exactly".

Applicn of this SOY "soln" to Cover's portfolio:

Say $\vec{a}_i$ — n vectors, for n stocks — are successive yield vectors so $z_i = \alpha_i + 1$: α_i w.o.l.

We want wts. $\vec{b} \ni \prod_{i=1}^{N} (\vec{a}_i \cdot \vec{b}) = \max$: call the soln. $\vec{b}^*$. How much SOY in is $\vec{b}^*$?

$\vec{b}$ is determined to be of total length 1

$\sum_{j=1}^{n} b_j^* = 1$
all components ≥ 0

For uncorrelated, successive $\vec{a}_i$'s: we can get a daily "sample" of the d.f. for all N days!

The mean yield will be $\nu \frac{1}{N} \sum \vec{a}_i \equiv \bar{\vec{a}}$

So we use $\vec{a}_i - \frac{1}{\sqrt{N}} \cdot \bar{\vec{a}}$ as the "daily sample".

The "best" daily choice $\vec{b}_i^*$ will be a $\vec{b}_i^*$ with a "1" component for the j for which a_i^j is max.

For "sample chunks" a few days long, the $\vec{b}^*$ value will also have all zeros & a single 1 for the component w. best (max yield for those days, because the cross products are too small to have much effect.

The result is that the daily samples don't even look much like the yrly or "5 yr"'ly samples.

How large must N be before the cross product becomes important? — consider only linear & quadratic terms, to start out:

$N=1$ $\quad \sum_n b^n (1 + \alpha_i^n) = 1 + \sum_n b^n \alpha_i^n$

$N=2$ $\quad \sum\sum (1 + \sum_n b^n \alpha_1^n)(1 + \sum_n b^n \alpha_2^n) = 1 + \sum b^n (\alpha_1^n + \alpha_2^n)$ — linear terms.
$+ \sum_{n=1}^{} \sum_{j=1}^{} b^n b^j \alpha_1^n \alpha_2^j$: The j,n quad component is $\alpha^j \alpha^n$

.39

→170.01→

131.90
91.161.90

On Letter to σ: — t. Altzheimer problem.

The most apparent diffy is the system's inability to remember anything for a long time. If this is fixed ---- (and it probably can be, thru ↓ of proby limit to ϕ or some small value — or some other way), Then, you should show that there exist TS φ's that could bring it to discover impt. heuristics. —

As it is initially, it is, heuristicly speaking, Very Bad: It discards immediately any idea that doesn't bring quick "returns" (Hyper greedy algm): it has no way to learn anything positive from what appear to be failures.

Mention causally that our goals may differ somewhat: But not say just how! If he asks: explain that I'm interested in devising U.S intellectual tool: That creating an very intelligent & living creature is quite another thing ---. I'd like to hold it as possl., but I really don't yet know how.

~~As it is,~~ I have found the system difficult to analyse from a purely theoretical point of view. T. system is in 2 parts: The probability distribution, P, and the working memory, W. P evolves by a pgm that involves both P & W; W evolves by a pgm that involves both P & W.

[SN] Suppose t. system wanted to test a cond. that had hy σ in its yield, Tho for long runs, t. system would do fine / But it would have hy proby of being discarded (permanently), early in t. trial. σ's system could deal w. this by trying it again & again until it got over its initial period of danger.

9:10 A 83°(Hg.)
9:34 90° Hg!

[SN] Re: The "horizon" effect: Say one had a very smart automobile! that would take one anywhere one liked. Upon reaching its destination, one gives it an "R" score telling how satisfied one was with this trip and destination. The machine also has other inputs, like my conversations it hears and my blood pressure during the trip.

From those inputs to the car, and many trials, it would eventually figure out where I wanted to go and acceptable means for getting there — but the training would take a long time and be very wearing on me. It would be a lot easier on the machine and on me if I were able to tell the machine directly where I wanted to go, rather than have it learn this from various clues in my behavior.

In a similar way, if I had a very intelligent "machine assistant" that I gave many research problems, I would like to be able to tell it the time horizon of interest, rather than have it induce this value from my pattern of reenforcement. Even if the machine were very intelligent, it would take a long time for it to reduce this "horizon" value, and when it did, I'm not at all sure it would get anywhere near what I wanted. | would induce

---

Many yrs ago at the Dartmouth conference, ~~We considered~~ someone asked ~~for question~~:

Q: If you had a very intelligent machine, would you ask it to give you what you really wanted — that which would please you most?

Several of us decided that we would never ask the machine ~~to do~~ to do any thing like that

a track: First of all, the Q. is very ambiguous: is the criterion immediate pleasure, or pleasure as viewed backward from a day a year or a decade's time?

A second, more ominous consideration: What we "really want" is very much in the "unconscious mind": — It's what we act "as if" we wanted. . . . .

~~This one~~ These unconscious desires can be very unreasonable ~~and~~ . . . not only socially unacceptable, but thoroughly unacceptable to one's conscious mind after a period of ~~time~~ considered afterthought.

Many of man's goals are ~~meant to be~~ used in guiding his behavior ~~not by~~ by moving toward the goal. The goal itself need not be achievable — in fact achieving the goal might be very counter productive. — The goal is for guiding — not for achieving. By aspiration. best

---

Main pts of Letter: 6) General Non-el goal is of much interest. (but not by me)

1) Poor long term means (probly tractable)

2) Gone seems not so good: Very "Greedy".

3) As is, system would take unacceptably long time to become very smart — unless it had suitable ~~t~~ sq's

4) Non-el Goals are usually best solved by ed. approxns.

5) His criticisms Q's & criticisms of my writings have be useful some:

I have extremely little useful feedback on this.

"I have had relatively little useful feedback on the Sol 86, 89 and so I much appreciate your comments & criticism.

7·15·97 ~~TM~~ LHL

6) That many of my comments ~~in~~ on the 1993 paper may be irrelevant in view of ~~recent~~ modifns. you've made since then!

I'd appreciate ~~copys~~ hard copys of .... and any other papers you think it might find interesting

7) comment on SM & ~~AMM~~ papers. ANN

8) The Horizon effect

(SN) Pattens in "problem Boundaries" & it, is part of a TSA.

7/22/97 TM Soy (SMA

$\times\left(1+\sum_k b^k \alpha_3^k\right)$

.01: 166.40 $N=3$: take all terms for $N=3$; adding $\sum_h b^h \alpha_3^h$; which gives all of the linear terms. The new quadratic terms are $\sum_j \sum_k b^j \alpha_3^j \cdot b^k (\alpha_1^k + \alpha_2^k)$

The total j,k component is $\alpha_1^j \alpha_2^k + \alpha_3^j \alpha_1^k + \alpha_3^j \alpha_2^k$: The new/quad components are α_3^j mult by the linear component of $N=3-1$.

So we have a simple eqn for the k^{th} linear component for $N=N$. it's just $\sum_{i=1}^{N} \alpha_i^k$.

The $N+1^{st}$ quadratic component $= j,k$ th N^{th} quad component + (the N^{th} linear component) $\times \alpha_N^k$

so the j,k^{th} quad component is $\sum_h \alpha_h^j \left(\left(\sum_{\ell=1}^{N} \alpha_\ell^h\right) - \alpha_h^k\right)$

$$= \left(\sum_{h=1}^{N}\sum_{\ell=1}^{N} \alpha_h^j \alpha_\ell^k\right) - \sum_{h=1}^{N} \alpha_h^j \cdot \alpha_h^k$$

(20) ? $= \sum_{k=1}^{N} \alpha_k^j \cdot \sum_{\ell=1}^{N} \alpha_\ell^k - \sum_{h=1}^{N} \alpha_h^j \alpha_h^k$. } is this correct for $N=2$? compare: 166.39 - .40.

This is very confusing, But it should end up w. a simple result.

The result for sample chunk length of 1 day: If one of the stocks has a mean (linear) yield much larger than the others ("much larger w.r.t. the σ yrly of that stock), then that stock will very often be chosen as the "daily peak", & we end up w. small Soy (?!).
This is true in General, & not narrowly for Covers System.

T result (.20) seems o.k.

Say $M_j = \sum_{k=1}^{N} \alpha_k^j$: $M_{ij} = \sum_{k=1}^{N} \alpha_k^i \alpha_k^j$ = cov. matrix.

= total yield for stock j

(.20) = $M_i M_j - M_{ij}$. It is interesting that the cov. matrix comes out w. a negative sign!.

If N is large, is $M_i M_j \gg M_{ij}$?

If M_i & M_j have any mean drifts, this drift will be $\propto N$. so $M_i M_j \propto N^2$
M_{ij} will be $\propto N$, however, so for large N, M_{ij} is negligible, wrt. $M_i M_j$.

.01 Re: **Soy**: Suppose we have 100 [rsl / strategies w.] indiv/ Gaussn pd's of mean 0, σ=1. Added to the bunch! One [strategy] of mean .1 & σ=0. We take ssz of 10, say: ▬
Which is the best to bet on? — How good is it?

If we considered the first 100 via the Soy method, we'd find an expected yield of 0. — so we'd bet out. last strat w. mean .1.

However, if we use soy on the 101 strats, I think we end up w. expected mean of near

.07 zero! — so "Paradox" — "inconsi". → [See 201.01ff for more detaild analysis]

This Seems to "REVIVE" the SOY Problem!

The 100 indiv Gaussns [samples] each has its own mean & var. The means are normally distributed about zero. The single strat of σ=0 & μ=.1 would not much be noticed by the SOY soln, I've gotten up to now! After Soy correction, the expected yield would be close to zero, w. some large varc.

I may have to Go back to the less el. approach of 147.11–.14

.20 On the other hand, this "current Soy "soln" does answer a Q that one mite often ask. What .01 ff says, is that this business of selecting the apparently best strategy is sometimes not the best way to proceeede!

.24 The least el. way to proceed, is, essentially 147.11–.14: i.e. we ask, not "What is the best strat to use"? But given the evidence and these models, what is the best way to bet in the future?

The problem is still "up in the air": I really don't clearly understand it!

[9.6.97] → 200.01ff is a reexamination, ~~[illegible]~~ — the idea of most of ▬ being that (.01–.07)ff isn't a clear difty ~~[illegible]~~ — However, I think 202.23 may be the true soln. — it is the approach I used earlier in the original "SMA" problem. [200.01–202.16]

Starting W95
→ Pix tool
→ Desktop → ???

10:35 AM
Max 122.16 + 1.67

As the years scroll by
The birthdays are like
the ticks of a clock!

Next wk
need ????Mann

Recomp

See 200.01

7/30/97 TM : Coding Theory ; SM :

Use of a/ deriv of a SM strategy & its yield, as a deriv. of a corpus.

[Many previous refs : 91SM 38-7.01 to 38-8.40 0/30/91 is one early ref.]

The strat & yield deriv. is a constraint on the corpus, since only certain corpi could have that yield if bet on in that way.

.05 A strategy is any algm for betting : One fairly general form : ~~[illegible]~~

.07 $f(t)$ = a function of the corpus before t: It tells what fraction of one's fortune is/to be made long or short, at time, T.

To the extent that a strat/yield pair cut out a subset of possible corpi, then the area of that region of corpus space measures the utility of that strat/yield pair & i.e. measures the expected yield of it (if properly used for betting. If the system is of the form .05 – .07, it may already be in optimum betting form, so its/expected yield is max expected yield from that strat, yield deriv.

The "precision" needed for the params of the strat & the yield !

.18 Say R is the k dim space of strat. deriv. & yield. Each point of that space defines a region of corpus space.

* density of default d.f. = (capital func?

One possible approach : one has this particular corpus, that's in a space one has a default (capital) on. Each pt. in R space of .18 defines a region of corpus space containing the corpus. The smaller the space is, the less info needed to specify the corpus (to a certain "precision" — so the corpus occupies a certain k volume in R space; say it is ϵ. Then ~~If we specify that~~ if a pt. in R space specifies a region in R space (containing ϵ) having the volume S, then the pc necy to specify ϵ exactly is just $\frac{\epsilon}{S}$. (So we want S to be small — as close to ϵ as possible.)

For each pt. in R space that defines the corpus (there is an entire subspace of R that does this), we have a corresponding S value in corpus space.

An entire deriv of ϵ, will then be $\int_{\text{(this subspace of R)}} \frac{\epsilon}{S}$ dR space

In some cases, the integrand will be ~~[illegible]~~ peaked over a small region of R space — i.e. $\frac{1}{S}$ will be sharply peaked as a funct. of points in R space & one can approximately integrate $\frac{1}{S}$ over that region. The peak in R space will be the pt. at which S is smallest.

So if D is the apriori density function in corpus space, we want the pt. in R space at which $\frac{D}{S}$ is max ; (or $\frac{S}{D}$ is $\min$)

Is this related in any way to picking a [illegible] set of strategy params

.37 of my yield? → 172.06

My impression that if a strategy is derived as of .05 – .07, then, if the/total code p cost is > the default pcost of the corpus, by a factor f, then one can gain a

7/30/97 TM : SM :

coding theory

Factor f by betty, if there is no Brotherage.

I think this latest foray on this strategy coding problem may be simpler & clearer & more usable than my previous works; (But I've forgotten details of previous work; So try to find some more recent refs.

I may have done some early work in Stanford (~1990, 1991),

.OG: 172.37: Why ask for/a hy yield? → ptr in stat/yield space — well The hy yield direction in R_+ space is (perhaps always) a direction of lower density in corpus space. Is this (usually/always) true?

One version why hy yield must correspond to small region in corpus space: Integral over ln. of all yields must be 1 (or something like that).

8597 TM LHL

2.6¢ cu = 1 penny
83¢/lb.
1 lb = 456¢;

$\therefore \frac{83}{456} \times 2.6 = .474$ (1)

Zn = 72¢/lb.

Dear Jerry:

In the past few months, I've ~~found time~~ been able to look at some of your writings?. — Some impressions!

On "learning how to Learn" (Jan 95): Your remarks about the very general learning problem are well taken. The problems in the ~~real~~ "Real World" are not nicely divided into sub problems. We are never given the same problem to solve. If there is a goal, it is usually not confined to activity over a short period of time. ~~but~~ ~~with respect to~~ what we do over a long ~~time~~ period of time. By starting with very little "bias", yet keeping Problem structure set universal, you make it possible for the machine to transcend ~~any blind spots in the~~ intellect of its designers, the intellectual inadequacies of its designers.

$\frac{72 \times 2.6}{456} = .41¢$

$\frac{83 \times 3.5}{456} = .64¢$

.04

SN Best just to write this letter. Don't try to make it perfect. Then Continue writing & perhaps rewriting until it says what I want to say.

~~It is my impression that most researchers think that those~~ workers in machine learning are not aware of the disparity between the problems they choose to work on and the problems ~~faced by a~~ normally faced by a living creature in a real world.

Associated with this lack of bias and universal instruction set, is a great penalty — it takes a very long training sequence and much computation for the machine to become very bright.

Organic evolution starts with relatively little bias, and a complete set of instructions, but takes a billion years of intense computation before it begins to solve very interesting problems. Artificial life starts with ~~a set~~ an instruction set chosen more appropriate for the problems it solves — and a much simpler set of problems than are faced by the creatures being designed by organic evolution. It takes A life much less time ~~than a billion years~~ to solve its problems.

~~In this approach~~ The usual way we deal with the very general, very realistic problem you propose, is to simplify it in various ways. — make it much easier for the system to solve. Some ways of ~~the~~ highly possible simplifications,

1. We divide large problems into smaller, clearly defined sub-problems, before we give them to the system to solve. We may allow the system to propose many trial solutions for a particular fixed problem, and give it a score for each proposed solution, telling how ~~close to the ideal it was~~ good it was.

After the machine has learned to work problems of these kinds, we give it more difficult problems: i.e. related problems that are more difficult — that are progressively closer to the kinds of problems that occur in the real world.

3.0¢

.01 The learning ~~sys~~ system you propose is (perhaps with a few modifications which I will discuss later) probably "Universal", in the sense that by using a suitable training environment, we can bring it to any specified level of ~~Problem~~ ~~Solving~~ performance in ~~problem~~ problem solving or whatever else we desire.

(insert 1.09)

.04

.05 ~~Please~~ It is certainly not our purpose in this — There are many substantially apparently "weaker" systems with this capability.

SN: How can .05 be (demand? shown?): E.g. consider a system w. problems given in a fixed way (w. spatial & temporal boundaries for problems) & a R level given for each proposed soln.

---

13 : 1.40 What has been done here is ~~to show~~ the final goal, of having a machine capable of dealing with real world problems and ~~providing~~ devising some subgoals for the machine. ~~In doing~~ This enables us ~~to speed up the machines search for~~ to speed up the machines search for solutions by an enormous factor — ~~but~~ but we of necessity ~~may~~ bias the machines search for a solution — and we make it easier ~~will usually encourage it~~ to find ~~suboptimum~~ what are probably not the very best kinds of solutions.

TI Memory loss problem: May be have a very large (potentially infinite) register where it can only write once & read ~~many~~ 8 to 4

There are other ways to help the machine ~~to find~~ find solutions with acceptable computation and training cost — but they all have the effect of limiting the machines search ~~and probably causing it to~~ so they do not consider certain very good solution techniques that (we, the designers, ~~you and I~~) are not aware of.

There is a necessary trade off between training ~~acquiring~~ a machine ~~that is very~~ to be very skilled in learning of all kinds, and ~~particular having~~ — considering an acceptable training cost — both sample size and computational resources. Another part of the tradeoff is how much of our own personal biases we ~~make~~ give the machine by selecting its instruction set, by selecting problems for it, and otherwise supervising its education. Each of these factors (& intelligent consideration of) can reduce training costs considerably, but can also introduce biases that ~~prevent~~ preconceive from acquiring optimum learning skills.

Ratio

Zin 7.14 / 8.92 = .8

2.5 / 3.6 = .83 = .84

2.1 ✓

3.6 ✓

8887 TM LHL

.01: Posed, insert following 1.40!

2. (1.40) The better Engineering schools follow a similar program: they start with simple, theoretical problems, then work on progressively more realistic problems — until at a certain point, they are working on problems that occur in the real world of engineering. Often they are not presented with more problems, i.e., deciding what problems to work on toward a very ill defined goal .... but some of the students are eventually able to acquire that skill (not necessarily through anything learned at school). → 2.13.

7k

to single-handedly deal with the full impact of the real world as adults.

1 Mol 3nt 20hr.

$ti = E \cdot C \qquad t = \frac{20\,hr \cdot 3\,nt}{nt} = 60''$

30 days + 50

.13 After a kitten is 2 or 3 weeks old, the mother cat

A mother cat keeps her kittens in a very friendly environment for several weeks after it is born. Then gradually she leads the child out into the real world, to fend for itself, as an adult — where it usually learns how skills it mother never taught it.

In other animals, the period of parental supervision may be much longer. Human parents often take as much as 20 years before releasing their offsprings to the full impact of the real world, as adults.

In intelligent systems one has a similar "adult" threshold. Before that time we are very careful in training the systems. But after adulthood is reached, training is progressively easier. This level might correspond to its being able to read a human language and understand enough of it, so that it will understand even more upon reading more. A lower threshold might be one in which it is not too difficult to modify human textbooks so that the machine can understand them.

My direct goal is not to design and train a machine to work that kind of real world problems that your system is aimed towards. I simply want to get the machine to an adult level as soon as possible, then try to devise training environments for it that enable it to solve progressively more difficult problems. I presume that this could be relatively easy after it had either reached a suitable "adult" status.

I think Douglas Lenat feels that his "Eyc" system is close to "adulthood" — but I am very skeptical of his performance. It seems to me that this is unlikely, but I have not studied "Eyc" to have a serious opinion (I am not familiar enough with

[illegible]

Items to discuss:

1) Some summary of <main "hylevel" pts in early part of the latter>.

2) relative amt of time spent on main problem vs. "self improvement" defn. of "self-improvement". The 50% ratio is o.k. — but, in fact, this is not an impt. problem. At a certain pt., I will prob'y ~~do to~~ allocate 100% to s.i. Also, for t. machine to learn ~~[illegible]~~ to optz this ratio, would take too long.

3) ~~a~~ ~~[illegible]~~ Time sharing method of Lerch' can be > twice as fast as method I dcrbd. — but uses much more ~~[illegible]~~ memory.

≠ 4) The ~~[illegible]~~ "lossy" many problems in his system; two possible ~~[illegible]~~ solns.

5) The example runs only used ~ 1/12 of usable time. (Check on "1/12").

6) I'm not at ~~all~~ ~~[illegible]~~ all sure t. general ~~[illegible]~~ system he proposes is ~~[illegible]~~ ~~[illegible]~~ good. ~~It is diff~~ I find it diff (so far) to analyze from a theoretical pt. of view. However, it is suggestive, and I may find it to be a useful beginning for a system somewhat like it.

7) On t. "s.imp" defn. How it involves "T" or approx. — why I'd not want t. machine to have to learn "T" (or approx.)

SN ⊙ What about learning hours? Is this "self improvement?".

s.i. includes modifying t. params of t. P.D. after a problem has been solved. "direct" problem solving" is soln of ~~existing~~ problems using Lerch's existing P.d, s.i. involves (at first mainly) modifying t. P.d. ~~[illegible]~~ partly in view of new recent info on solved problems. Later on, optzn wrt. more Global Goals, will be implemented. T. ~~[illegible]~~ more Global Goal: TM watches TM1 do Lerches on problems! ~~Then~~ Then tries to modify p.d. so as to ↑ expected Gorc in future.

.32 [5 Jun 88? Solomonoff] On t. hy level "TM2's Gorc": exponential wtd (future "expected" horizon") ~~[illegible]~~ R levels. This seems closely related to t. $M problem of evaluating future yields of an "action algm". $

Consider CB = ∞, CB = short; CB = medium. Also look at ruff approx. methods. How does the "problem pool" approach fit in? — It mite be a v.g. "study" ~~try~~ problem for t. Gorc of .32

Perhaps try Jaugau's Gorc on some "Big Problems"

((→ Possibly what J. had in mind was t. "anytime algm" — That if one stops t. machine at any time it tends to have a total max Gorc.))

{ well not so smart! T. soln. depends much on one's expc estimate of expected sampling frequency!

— But what about t. Levin search OZ soln. w. timesharing? It is a kind of soln. to t. "anytime algm". It seems to assume a very hy frequency sampling ? (?)

So TM_2 is watching TM_1 solve problems! [It knows TM_1's algm in detail. From this corpus of data, it is supposed to "improve" TM_1's algm. Actually, what TM_2 has observed: Various $(prob_i, TM_1^i)$ pairs & how they evolved.

If we simplify TM_2's Gorc, so that t. problem is to modify TM_1 so its likely that it would have hy $\overline{R}$ on t. past problems! This seems like a well defined OZ prob. — T. uncertain part is if (as is often t. case), There is not nearly enough time to test proposed Mods of TM_1 on t. entire past corpus. — In which case, t. problem becomes ∪ to requiring TM_1's algm to be good on t. future corpus. TM_1 can be [redacted] completely "opaque" to TM_2, or have different degrees of openness to TM_2.

(A) How people solve such problems: They make 1 or more models of TM_1's activities — that give relations of various params of TM_1, to its yield in t. past — Then try to find good sets of params of hy yield. —

So TM_2 is (always) working w. this Model, & TM_2 doesn't really know how accurate a. model is. A similar problem occurs in dctn of RW by means of (digital) models. In both cases, we don't know how much time to spend on finding a v.g. model.

(B) A superficially different direction of ["improvement" of TM_1's]: Say an algm. used by TM_1 can be improved in a uniformly better way. By doing this "improvement" TM_1 is guaranteed to do t. same problems as fast or faster (w. no other γ cost)

What other ways other than (A) & (B)

So: is this a generally occurring difficulty?: That we make some model of a phenomenon, Then we solve probs wrt that model. How close t. model is to t. phenomenon is sometimes adjustable — by using more cc we can get closer — sometimes we know how much closer — sometimes we don't know how much closer. The problem of dealing w.t. model can often be a well defined INV. or OZ problem.

.30 Often we "assume" certain laws of physics or apparent laws & define problems in terms of them.

In the "classical" TM_2 Gorc of 4.32 we could devise a model for t. future problems, Then we get a well defined OZ problem for TM_2 — Th. amt. of cc to spend on t. model here is impt., & quite unclear.

— In t. cases of (.30) (Models in Physics), it usually a lot easier. In "Social sciences" problems, t. amt. of cc to spend on t. model is also very impt. & very unclear.

Fortune — M. Barry

So: Say we did This! We make a model for future (or whatever area we need to model). We solve resultant OZ problem, w. C.B. = coz_0. We then improve our model by adding cm_1 of cc: we then solve resultant OZ prob. w. $cc = coz_1$.

Well, if we set $cm_i \simeq coz_i$; perhaps we'd not be off by factor of 2!? We could just go back & forth between $cm_i, coz_i, cm_{i+1}, coz_{i+1}, cm_{i+2}, coz_{i+2} \ldots$. Say here $cm_i = coz_i = 2 * cm_{i-1}$ so we double time.

Alternatively we could do it time sharing — i.e. continuously work on improving model! (cm_i) We then spend amount of time at coz_i = time spent on Model$_i$. . . . .

We obtain various models sequentially: perhaps Model$_i$ occurs (by definition) at $\sum cc = \gamma_0 \cdot 2^i$.

Well, Applying the "SOG Soln" to more situations in which a general understanding &/or Theoretical Soln is not available .... This has limits — say one had 1000 of such items & decided to use $\frac{1}{1000}$ of one's cc on each!

So far, however, I may only have 4 such items!

(1) Direct problems Given to TM$_1$: Both OZ & ENV. Problems.

(2) Making a model for TM$_2$'s Govc — involving outlining likely forms of future problems. (an OZ problem)

(3) Using Modifying form of TM$_1$ so it is as good as possible. (2) (an OZ problem)

(4) Mapping Real analog world into digital dcms: (This item is a bit different from 1,2,3)

What aspects of R.W. are relevant? Impt. in physics, but more obviously relevant in Psychol & Social Sciences.

J's Govc is like ordinary G.A. — very wasteful, inefficient — creating many trials that are illegal (syntactically n.g.). Koza improved G.A. by using potentially universal languages in which a much higher fraction of trials were at least syntactically correct (= meaningful). J's lang. is still old G.A. in its small fraction of progs that are meaningful or have much chance of being any good.

This lang. could be much improved by using any "functional lang" or maybe even just LISP.

SN Perhaps modify FORTH slightly so it is a complete "functional lang". At present, Forth can do compositions of functions & Loops: It needs be modified so it can use functionals on functions to get General recursive functions efficiently. — In particular, it should be able to express prim. rec. funcls very efficiently.

8/11/97 TM LHL

On man's "top Goals": Before agriculture was invented, sweetness and fatness were good guides to healthy acquisition of needed calories as well as other nutrients. So man "liked" sweets and fat. — They guided him to healthy nutrition. After intention of Agriculture, acquiring sweets and fat became very easy, so much so, that they were no longer a guide to good nutrition. These factors were no longer a guide

Another desirable behavior in many early societies is for male neighbor to destroy their wives, and property and ability to reproduce themselves more extensively. acquiring With modern weapons, it becomes possible to kill as many people as one likes, and this particular goal becomes counter productive for the society.

[The general drift of this idea is to show that a man's present day goals could become extremely inappropriate were he to have access to a very high level TM.]

Another approach: That I'd worked out problem of unified Core for TM. (probably spent too much time on it, since it is not a bottleneck) — But 1. soln. has purposes in one that I'd never considered:

reason: In early stages of life, system throws away or moves into — into on suboptimal trials. Ordinarily info about what did not work is about as useful as info about what did work.

At a later state of development, system could choose to store all info about its past trials and their successes or lack of success. By not making that However, it may take the machine a long time to get to this stage of development — and before it does, it will be considerably handicapped.

Another item I'd thot about a lot but for which I've used a quite distant approach: The future horizon of the System. The goal of the system is to maximize the total "R" value that it obtains during its life. It does not, however, know how long it will live. If it did, indeed, know how long it had to live, its mode of activity would depend very much on just how much time it knew it had available.

Not having this information, it has to either induce it from the Reinforcement data, or somehow devise an optimum strategy that is independent of

5) Try to limit each discussn. to reasonable length.

6) Specific "~ Bugs" in his system: (a) Memory (b) Garbage collecting may or may not be adequate: We could avg. manl system so its easier for it to keep all old info; write ~ permanently. (c) (Bug?) only uses ~ 10% of time. (d) unsatisfactory computer lang: Good/Bet functional lang may be better

7) Hvr, in design: Better get system to try to work or try to work some problems, than modify system [redacted] to facilitate certain very impl. opt. techniques.

8) [redacted] #1 goal is to get lots of info into "P.D.": This is done by giving TM INV. probs & solving OZ probs using (non-optimum) FNV method. [redacted] Perhaps use a set of "fixed" optzn methods — one of these is the one I use for "compressing a P.D." after each INV prob. soln. is obtained. I can perhaps use some technique to find heuristics: i.e. regs in t. "traces" that lead to cheap solns.

.17 9) After a problem has been solved, we want to get several things from t. soln: (a) Soln. itself & how it maps to problem dcrn. so we can try to solve it w/ parts in future (b) [struck out] factorizn. of problem: its parts & its structure. Both parts & structure are given P.C.S. Soln. is "product" of "structure" & "parts".

1) System doesn't learn from errors
2) System is very "Goody"

.22 10) When something is put into "Memory" we must also designate how & when it is to be used; what ideas it is related to & how much, etc. This is true for both humans & Machines.

Also e.g. for problem solns, Put into memory .17–.22

In general, t. more there is in memory, t. harder it is to put new stuff in, since there are more things for it to be related to.

11) On "P.D.'s": Ts. "P.d." is a conditional P.d.: So if we're looking for a soln to an Inv. problem, it is conditional on t. dcrn of t. Inv prob. If we are looking for a soln. to a OZ prob. t. "same" P.D. is conditional on t. dcrn. of t. OZ problem.

In early life of T.M. Th. P.D. will be essentially unconditional, since t. machine hasn't had enuf experience to build up useful "conditionals". In t. case of OZ probs, t. P.d. is on t. set of possl. hill-climbing techniques.

When TM puts things in Memory, it (among other things) trys to find good conditions under which those things are to be retrieved.

The Conditional p.d. is [redacted] operator induction: we have, as corpus, $[I_i, O_i]$ pairs: In this case, [problem, soln] pairs. TM tries to devise a problistic operator to map any (new or old) I_j into a p.d. on all possl. outputs. T. corpus may include both INV & OZ problems.

81397 TM LHL

this [redacted] information.

In organic evolution, the problem is quite different from the one you've addressed. In a particular environment, many organisms can be tried — each ~~nothing~~ one acting as if it expected to live a different amount of time (Tau). The ones with the best Tau's would tend to get higher scores at the ends of their lives.

In the problem you describe, there is only one life — [redacted] I expect that the problem of [redacted] inducing a good value of ~~E~~ Tau from the pattern of reinforcement is a very difficult problem [redacted] when it is solvable at all. I suspect that for many reinforcement algorithms, the ~~problem is not solvable~~ needed information is not present and the system ends up with a value of ~~Ed~~ Tau that is very small.

For the ~~x~~ kinds of systems in which I'm most interested, I would rather have Tau be [redacted] a parameter that the user inserts into the system — as part of the description of the machine's set of problems.

Things are ~~modest~~ (or a bit in a certain order: I think these could be!

1) A large no. of probs. w. L srch as sole soln. method. ~~I~~ I. modify "p.d." to summarize meta/rep. problems, only. Shortw. I.M.V. probs.

2) TM learns to work OZ probs. like soft. ~~z~~ sys. pgms (not itself) from OZ probs. & sim probs.

3) TM is allowed to work on self.

4) TM may be allowed to try to induce horizon Tau

5) Also "later" (a) to divide large probs into smaller probs. (b) to put info from RW into digital form.

R. Rucker
Artif. life lab
= Both + S.W.

doc103.htm ~~z~~ = !

hep. CLAq's by — Gauss)

1P Cx (20!)
46.23 −.06
20.60 −.28
23.79 +.07
18.61 +.23

Science 8 Aug 97 p823

019 → 074

287278.36
1536003

.365 > .1% fd.
6 ~.4

.1

Aug 9: Marvin's 70th Birthday

Poffito on Music. Learning

I don't know it on el. ~~etc~~!

2P [redacted] A new tack for the letter:

1) [redacted] I agree that his normal goals are what we ~~may~~ ~~really~~ probably want to ~~attain~~ ultimately achieve — that many workers in A.I. do not recognize these goals.

2) I feel, however, that the ~~best way~~ fastest way to achieve these goals is not to pursue them directly, but to ~~design and train a systems that are easier to train~~ to work toward easier (simpler) goals. Then, ~~when these~~ having achieved ~~these goals~~ them, modify these systems ~~systems, ~~which have been~~ ~~can be~~ ~~modified~~~~ (by x ~~y~~) to work toward more difficult (realistic / final / useful / practical) goals ~~of you wants~~ you propose.

3) Some Simplifications: a) division into problem (TM₁) & Metaproblem (TM₂). b) → single Gauss broadband pass / undefined time horizon / → well defined " ... c) Clear division into subproblems d) ~~worker~~ initial work on INV & OZ probs using INV technique.

4) Discn. of "Simplify" process / Good toy problems v.s. bad toys

So! What does this "new Global View" have to say about the Heur discovery Problem? —
→ detecting useful regys in the trace of TM operation & modifying TM (or just modifying the P.D.) to take advantage of these regys to give faster/better solns.

To Get ideas for this problem: look at various heurs; look at the data that should have given rise to them. Try to find lrng. techniques that could have detected & implemented those heurs.

N.B. If I can't conceive of a corpus that should have suggested the heur, then I cannot "justify" that heur.

The "Quick Abort" heur. is of particular interest since it may have to work not thru P.D. modifn. — It probly characterizes a general class of heur that doesn't work thru P.D. Modifn.

Back to the letter of 181.26 ff!

.13 Dear Jaurjen! I have enjoyed reading your "Learning how to Learn" paper (Jan 1995), and have found it to be very thought provoking — forcing me to state more ~~clearly my g~~ clearly the goals and methods of the system I had proposed in 1986 and 1989:

~~I agree [...] that the~~ The goals of the system you present are certainly more realistic than those of ~~[...]~~ most other work on machine learning. They are, however, very difficult to achieve, if persued in a direct manner — i.e. I think it would take a very long training sequence ~~of problems~~, and a very large amount of searching, before it could solve interesting problems. (See 413 "remark #9".)

My own approach has been to start the machine with easy (not necessarily realistic) problems, then build up to more and more realistic ones — much as is done in ~~[...]~~ training human engineers.

In one version of the system, I start out by giving the machine a carefully designed sequence of (inversion and optimization) problems of increasing difficulty. ~~[...]~~ It solves each of the problems, using Levin search (L search) based on the probability distribution that it has at the time the problem is presented. The time between problems (or during problems, using the time-share version of L-search) is spent "compressing the probability distribution". ~~[...]~~ By "compression" I mean finding better regularities ~~"compression" not to be taken literally.~~ ~~[...]~~ the amount of in the data. ~~"Since compression" is the best measure of~~ ~~[...]~~ how good a proposed regularity is. "Compression" achieved is used to tell how good a proposed regularity is. The new found regularities in the data are then used to modify (improve) the probability distribution to be used for subsequent L search.

After the machine has begun to show skill in working ~~on~~ certain kinds of optimization problem, I allocate it time to be spent on "self improvement":

This "self improvement" is in the direction of the highest level goal of ~~the~~ your "Learning how to Learn" system. There are, however, important differences. In your system the goal is to maximize the total ~~[illegible]~~ payoff over the life of the machine. — ~~[illegible]~~ Apriori the machine doesn't know how long it will live.

(SN) Have section of letter on "Remarks about his Remarks":

See 146.01, 130.22 (criticism of "finite horizon analysis")

.12 SN On exponentially wtd. R into the future; (or R window version)
T. "HORIZON" problem. Don: 10 58 AM 11 45

.13 For R window: Whenever there is a choice betw. doing (a) task w. immediate value, or (b) a task w. longer term value, that increases speed of soln. of future tasks; It's always best to do (b) first. ← No!

.15 — No! 129.40 gives conds under which one does self improvement at all.

→ If one does it at all one does it first.

Hvr, even this last need not be true. Certain tasks ↓ in value rapidly if

.18 done later (like self defense or general {avoiding/dealing} w. danger).

Anyway, the arg. of .13 — .15 is that one always does s.i. (self imp)

.21 Dart (s.i.) first. so if the future window is a moving window, one does nothing but self-improvement, & one never does immediately useful tasks! If s.i. is

Args v.s. (.21): (1) (.18) tasks that have to be done (soon / immediately) (2)

23 not warranted by the 129.40 criterion (3)

One way to deal w. the problem: Have T.M. programmed to work specifically on certain problems I give it. Between these problems it works on s.i. (This amounts to (.23) (1) (≡ .18)

30 → Another way (perhaps much less cl.) Give TM problems of ↑ difty, so that eventually probs are so difft, the search needed to take so long, that TM finds s.i. to be a ~~[illegible]~~ cost effective part of the soln.

Another way to deal w. .13 — .15: s.i. is always done in ∥ w. the main direct tasks. T. problem, then, is what fraction of Time share s.i. gets.

A serious difty, hvr, in some of the tougs. ! In general, no particular piece of work done by TM need be ~~[illegible]~~ labelable as "direct prob" v.s. "s.i."

(.30) certainly fixes this difty.

Drusrecant is in C!! ! BBLspecs.o takes 260MB! is in C!! ... special [illegible] But ISKY have Bablspace BSN

global credit assignment.

.01 In TM's original Solgy system, TM_1 used Lsrch exclusively: TM_2 would (only) by modifying TM_1's P.d., — T. idea being that all theorems would be expressable by modifns of the P.d. used in Lsrch.

Now: TM_2 has to have some goal for modifn. of T. P.d. (= "TM_2's Gone").

I originally felt this would be 1. exponentially wtd Σ future R (184.12) → i.e. objectives of 184.12ff (.21 in particular) — But now I'm not so sure this is a v.g. Goal!! 184.30 seems More attractive.!

Re. J's ∑ system: What its appeal is, is unclear: He presents a system: a PEM: What it would learn — what it is capable of doing, is unclear.

A theoretical difficulty w. J's system: The R(t) function can really be "anything": If R(t) (R is total recall.) is a function that gave good learning, then $(R(t))^{1-\epsilon}$ would probly not work as well — it would require that TM learn "faster & faster". However, we really don't know what R(t) is — we'd like it to be somehow linear in what was accomplished by TM. Using $R^{1-\epsilon}(t)$ means that eventually TM will have to completely "pop the stack" & start over. — It corresponds to a world in which a man is not much rewarded for behavior that on the average would be highly rewarded for. If the problems for TM do get much harder as time passes, then the $\frac{\Delta R}{\Delta T}$ > previous best $\frac{\Delta R}{\Delta T}$ can fail badly (186.08)

---

in the system of (.01), perhaps TM_2 tries to find a P.d. for TM, that (1) is of type (simple) (2) would have led to low cc Solns in the known past. [How (1) & (2) are to be combined is unclear!]

This is perhaps one aspect of the: Soln problem [Looks very much like "Soln problem" to find an "Action Algm" w. hy expected future yield.]

.27 Note also that TM_2 will usually not find it cc-efficacious to test all of TM_1's past problems on a newly proposed modifn. of the P.d. — This differs from my work (Phys for) on SM. (186.20)

SN Lang. lrng is fairly difficult. But I should be able to get TM to work Alg. (& perhaps Geometry) problems fairly well w.o. too much trouble. Getting it to learn Integration might be more interesting!

Note that solving linear, then Quad, then Cubic eqns was possible. It might be possible to get it to do quartic, then quintic (using Elliptic functions, ect)

My approach has been too narrow. Using hints/examples, as well as problems & other heuristic tricks should be tried. Any method usable on a human student usually can be adapted (sometimes w. much Modifn) to TM.

See how much of Polya's hours I can put into TM. — Also Zwicky

(& TM may already use much of Zwicky's tricks) — i.e. attempts at exhaustive search.

019 25 074 LH

8.21.97 TM [redacted] LHL

.01 → One Uncertainty: Just how to show all (normal) hours are expressable by modifns of t. "P.D."

.02 Well, first: Are all hours expressable as modifns of order in which t. cond. trials are made?
I think I've been assuming This.

.04 If .02 is true then: If one uses Lsrch, can any reordering of trials be expressd as modifn. of t. P.D.? — seems likely.

To help work on .02, .04, Make up examples of hours & how .02 & .04 apply (if they do!)
See 189.10–.18 for a bit of relevant discn.

.08 : 185.28 A Big Criticism of J: If his Goal is Max Σ R in life of machine — (Machine is of unknown life time, a priori), it is not clear that rejecting sup's w. < peak $\frac{dR}{dT}$ will do it — certainly not immediately — and it's not clear

.12 that it will do it in t. "long run". → 188.27

SN [redacted] "ALife" approach to A.I.: The Model is Sturgeon's "Microcosmic God":

I think t. critical Q is: What is t. reinforcement structure?
Perhaps reread t. story: It may be in "... in Time & Space" early Analogy (mainly Aspt)
T. details in t. story could be very suggestive. Sturgeon was a very brilliant guy!
Maybe ask Minsky!

.19 → 187.09

.20 : 185.28 Note t. difficulty of 185.27–.28 is similar to t. problem of finding an Action Algm w. hy future yield. I.e. in testing AA (= Action Algm) on t. past, one can just sample t. past. One leaves out data in t. past — which is ~ to "looking out" data int. future! ("~" but not t. same! Future is diffrnt from random selection of data to omit from t. past corpus!).

Dat.

Note: In t. "Soy" problem there is probably an implied cc limitation — but this is on making & testing models ···· presumably, using all of past data. In t. present problem, we can also save cc by considering not all of t. data.
T. SM & TM problems are extremely ~. Are there any diffrncess that might suggest "diffrnt ways to look at t. problem"?

Say one "considers" all possl. strategies: One finds from Least error on t. known corpus we obtain a set of seats: each has its pc, its cc, & its G. Certain A.H. starts of low pc. will have very hy G. What's a good estimate of future G?

[redacted] some ways t. TM problem may be significantly diffrnt from SM:
1) SM has a "default" corpus.
2) In TM, if t. past corpus was a soy, or IBM probs, we can often generate a sort of possl. "future Inv probs. that are ~ same as those of t. corpus, but w. diffrnt params!
So many A.H. solns to t. corpus problems would not work for this new "pseudo corpus".

.37 → What about picking t. AA w. max pc. x G? (& keeping "insurance").
→ Look at that work on preceding Soy Prob I did on SM/TM w. Tom last few Months.
I tried to do a "Review" of it.

→ 187.01
188.01

Also: work out t. Gorc for the "problem pool" form of TM! This is a somewhat different view & may be helpful! (Also it may be a better form for TM!)

So 3 problems: 1) SM & Soy 2) TM w. attempts to get Good AA for future.
3) TM's Gorc for t. "problem pool" form of TM.

. . . . .

SN Often when one has an idea on how to improve an AA, t. improvement is "Uniformly Good" — so one need not do empirical tests before adopting it: How does this fit in w. my previous ideas on "AA improvement" (≡ s.i.)?

09 : 196.19 "Micro cosmic God" My impressions: That the race of people developed would be very small — maybe developed from Microbes. The "God" would present problems to t. race. If they were not solved: Big penalties: If solved, big rewards.

The "God" devised a suitable TSP. The race responded by (rewarding / penalizing) individuals for work (away from / toward) t. "God"-given goals. Sunday Sept 7 4–7 PM.

A fundamental Q about t. System: In Sturgeon's system, each person is "intelligent" — t. "race" consists of many such individuals who cooperate to do tasks given by "God".

A somewhat diff'nt (a "Soc. of Mind" Minsky) would have many individuals, each not so smart, but rather tightly coupled, w. same top goals, hvr.

Also another apparent diff'nce: in t. Sturg version, one could have mixing of genetic descns of individuals — (perhaps in subsets of individuals) — to develop certain skills in a sub-population).

In t Minsky version, The "mixing" of descns of individuals would tend to be very restricted if it was allowed at all. (& Mm — If not allowed at all — how do we get long?!).

Anyway, the Sturg. version contains much more hardware in each "person" is "complete" & so, has to have a fairly large pgm. It would seem that there is much duplication & perhaps waste.

On t. other hand, training Minsky's system may be much harder.

In St. system: One could start w. an ordinary A life pgm: Then, somehow, we have to get t. individuals to form a social Group to work toward a common Goal — (i.e. pleasing God).

8.23.97 TM LHL. T. HORIZON size Problem (.27)

.01: 186.40: Re: 186.37 (Max R.C. vs G): Soy is not so much concerned w. future as it is w. knowing t. expected future G. 186.37 is concerned only w. picking t. "Best bet" of all possl. AA's.

[[One of t. disturbing outgroupings of t. "Soy" investigation was that precamy t. apparently best strategy was not always t. best thing to do! (171.01, .20)

Hwr., I'm not sure I understand this — example is I'm not sure my Conclusion is correct.

Another (possibly) disturbing thing about 186.37: It doesn't seem to take into account t. "width" of a peak — (in t. continuous case) or many near by" but not exactly t. same, saints, in t. discrete case. E.g., in t. case of neural nets "flat peaks" are best.

Does t. "Flat peak" idea assume some sort of "closeness function" betw. t. "saints"?

Make a list of so impt. open problems in this area that seem closely related..
—From SM & TM.

.13 In case of N. Nets: "Flat peak"! T. peak is wtd. sum of models. Hwr., if we have to pick a single model, t. center of a broad peak is ≈ same as sum of models in t. peak — so it's a good approxn.

→ How to apply to 186.37 is unclear! ⇒ for t. continuous case maybe a w. argt.

.19 is possl! → 200.10

.20 T. argt. of .13 is same for Cover's Univl. Portfolios — so it would justify Cover's method. In t. case of Cover, since t. AA's are all linear, the combined method is one of the methods — not exactly t. peak, but close to it. His claims of how good it is may be an effective applicn. of SOY. ... (But why does he get such a pessimistic estimate?) (i.e., low future exponential yield).

That "this peak" is good is a local average so width is impt: Well: in Both ANN & Cover's Univ. Port., t. AA is p.c. or "like" pc, so perhaps this is why wtd sum is correct. (in ANN, t. output is p.s.).

.26

.27: 186.12 SN On "Not knowing how long it will live": First, it acts as if it would live for T years. If it does live for T yrs, it then acts as if it will live 2T years, &. If it lives for $2^N T$ yrs, then it begins acting as if it will live for $2^{N+1} T$ yrs. So, if it lives for time $T = 2^* T$, Then it acts as if it will live for time $2 \cdot T$.

While the TM of .27 might not be have exactly as desired, its a step toward a soln. It would have a reasonably hy $\frac{dR}{dT}$ during its life. In a TM that was "A scientist's tool", I think I'd want to control how much time TM spent directly on problem v.s. time spent on s.z'. I'd like detaild control: so I could tell it what fraction to spend on each, or To spend "all" of its time on a particular problem that I wanted solvd as soon as possl.

If TM did live a very long time, it would spend long continuous chunks of time on s.z'. w. no directly usable output. For a "scientist (helper tool assistant)" this would seem to be not very good!

→ T. fmly. dgcn. casts some Q on t. utility of J's goal of Maxing ΔR during (unknown) Lifetime of TM!

8.23.97 TM LHL

.05 Well, 188.27 clarifies t. "Time horizon" in my mind a lot: (along w. t. earlier discn of 129.01 ff, & 130.22, 146.01)

.03 Back to t. diffys of 188.13 – .26 : In TM, having a AA that gets solns. of past problems faster in all cases ("Uniformly") would be a clear improvement.
However, we want TM to extrapolate ① to problems "~" to those of t. past.
.06 ② to note trends in problems so it guesses what future probs mite be & what techniques
.07 would be valuable for them. ③ In t. "problem pool" type of TM: TM examines problems in t. pool & decide on what techniques have to be developed
.09 to solve them.

.10 In my earlier post on "s.i." I mainly considers s.i. of rufly 2 types:
.11 ① Simply to modify P.D. in a conventional statistical way (pretty by using problistic Grammars) so that t. P.C. of known corpus of problems was as large as I could get it.
.13 ② from t. "Traces" of past searches, modify t. P.D. to further ↓ total cc of searches of t. past.
It seems that these 2 problems were "well defined".

.18 Whether ② (.13) is possl. is t. q. of 186.01
If ① & ② (.11, .13) are possl. then they constitute a kind of el. "soln" to t. S.I. problem.
...... They tend not to have t. Breadth of .06 or .07, hvr.

My impressn. is that t. problem of .11 ① may be well defined: I'm not sure about .13 ② – (also t. q. of .18)

Re: .13: In .11 t. goal is to find values of params that derb. t. P.D. (so as) to maxz p.c. of solns. found.
" .13 " " " " " to minz t. cc of t. solns. found.
So they are both OZ probs, but w. difrnt Goals: we could do .11 first (it seems easier) then do .13 which is harder & has more input into, & is closer to our true goal.
t. goal of .13 is to minz. $\frac{\text{cc soln}}{\text{P.C. sol.}}$ | t. goal of .11 is mainly to maxz PC soln. } a more final goal (≈ "true goal")
In .13 t. goal may not be well defined because (again) of t. trade off betw. being (low p.c.) a.h. to obtain solns. of past problems & getting very low cc;
v.s. hy p.c., but not so low cc solns.

It looks like .13 is t. general AA optzn. problem again.)

.31 So again we have t. same problems of "is 186.37 correct?" & t. discn of 188.01 – .26; 189.03 – .09

It seems that in both t. TM & SM versions of t. problem, we are concerned only w. t. future & how "close" it is to t. past corpus & in what way(s) it differs from t. past corpus.

.36 For TM: / we give him t. eq. $3x^2+17x+4=0$ to solve;
Somehow it knows that t. generalizable params are 3, 17, 4, (perhaps 0), but t. exponent "2" on x is not so easily genzd! I.e. t. methods usd to solve .36 do not readily genz to $5x^3+2x+18=4$. (to some extent they do, —

8 24 97 TM:LHL

CUBIC EQUATIONS: soln.

If TM has learned how to solve linear & quadratic eqns properly, he can solve cubics w. "~" methods. . . . But I really don't remember how this worked! — see .37

$x = \beta y^2 + \alpha y + b$ (struck out)

$X^3 + AX^2 + BX + C = 0$

$X = Y + \alpha$

$Y^3 + 3\alpha Y^2 + 3\alpha^2 Y + \alpha^3 + A(Y^2 + 2\alpha Y + \alpha^2) + BY + B\alpha + C = 0$

$X = \alpha Y + \beta$

$X^3 + BX + C = 0$

$\alpha^3 Y^3 + 3\alpha^2 Y^2 \beta + 3\alpha Y \beta^2 + \beta^3$

$+ B\alpha Y + B\beta + C$

$\alpha^3 Y^3 + 3\alpha^2 Y^2 \beta + 3\alpha Y \beta^2 + \beta^3$

$A\alpha^2 Y^2 + 2A\alpha\beta Y + A\beta^2$

$+ B\alpha Y + B\beta = 0$

$X = \alpha Y + \beta$

$3\alpha^2\beta + A\alpha^2 = 0$ | $3\alpha\beta^2 + 2A\alpha\beta + B\alpha = 0$

$3\beta + A = 0$ | $3\beta^2 - 2A\beta + B = 0$ ← No!

$\therefore \beta = -\frac{A}{3}$

$X^2 + BX + C$; $X = Y + \alpha$; $Y^2 + 2Y\alpha + \alpha^2 + BY + B\alpha$; $2\alpha + B = 0$

$X = Y^2 + \alpha Y + \beta$; $Y = \alpha X^2 + \beta X + C$

$Y = AX^2 + BX$; $X^3 + Y = 0$; $X = \frac{-B \pm \sqrt{B^2 - 4A(-Y)}}{2A}$

$Y = \frac{\alpha}{X} + \beta X + \gamma$; $X = \frac{\alpha}{Y} + \beta Y$

$\frac{\alpha^3}{Y^3} + 3\frac{\alpha^2}{Y^2}\beta Y + 3\frac{\alpha}{Y}\beta^2 Y^2 + \beta^3 Y^3 + B\frac{\alpha}{Y} + B\beta Y + C = 0.$

$\frac{3\alpha^2}{Y}$ $\quad B\alpha + \ldots = 0$

$X^3 + BX + C = 0$ $\qquad X = \frac{\alpha}{Y} + \beta Y$

$\frac{\alpha^3}{Y^3} + 3\frac{\alpha^2}{Y^2}\beta Y + 3\frac{\alpha}{Y}\beta^2 Y^2 + \beta^3 Y^3 + \frac{B\alpha}{Y} + \beta B Y + C = 0$

$\left(\frac{3\alpha^2}{Y}\beta + B\alpha\right) Y \quad \left(3\alpha\beta^2 + B\alpha\right) = 0$

$= 0$ $\qquad \alpha^2\beta = \alpha\beta^2$ $\qquad 3\alpha^3 + \alpha^2 = 0$

$\alpha = \beta$ $\qquad 3\alpha + 1 = 0$

$X = Y + \frac{\alpha}{Y}$ $\qquad \alpha = \frac{1}{3} = \beta$

$Y^3 + \frac{3Y^2}{Y}\alpha + 3Y\frac{\alpha^2}{Y^2} + \frac{\alpha^3}{Y^3} + BY + \frac{B\alpha}{Y} + C = 0$

$3\alpha + B = 0$; $3\alpha^2 + B\alpha = 0$ so $\alpha = -3B$ solves it.

Phone 8 morn to 10 PM, (1-8 visiting)

238 1500

Benj. House -

Roxbury

120 Fisher Ave.

Brigham Circle

Murray Denotsky

.37 → [But surprising that this substitution should simultaneously remove 2 terms!!] ←

Go back to 1990 (or 91) See how I solved/(math) quad, cubic.

(SN) Barak Melzak Algebra! (Survey of) (pp 120 ff) Discuss of how to solve quadratic, cubic, quartic! TI explains seem to leave out essential hours. Could TM learn anything by looking at these solns? Are they useful "hints" on how to solve eqns?

d) (SN) Consider a corpus as a large (maybe all) set of AA's. Each has a (mean) Value wrt the corpus. By knowing all these AA's & their "values" one can often (maybe usually) reconstruct the corpus. If each AA is a component of the corpus, then clearly there is usually a lot into to reconstruct the corpus.

.05 • We may regard each AA, value pair as a partial coding of the corpus, since .06 it narrows down what the corpus can be.

Is "Coding w. a Fidelity Criterion" relevant? T. "Fid. Criterion" is somehow related to the Yield.

(SN) Note! Min $\frac{cc}{pc}$ is Min Cost: A noble goal (or -G) — an AA "value" (.01)

$pc \cdot \frac{cc}{pc} = pc \cdot (-G)$ so picking Max re Σ min $pc \frac{cc}{pc} =$ Min cc.

I feel that the AA's a. values near the peak are somehow very impt. re wrt future values of the central AA.

Perhaps I feel that the future corpus will be algorithmically near the past corpus, & that the ↓ in G's of [redacted] AA's, as they denote front. pack, is somehow related to ↓ of G of corpi near the (known) past corpus.

This idea could be perhaps made exact by ideas like (.05)

---

1) List some examples in which a soln. to the problem is possl. or a good approx is possl. (p.c. calcn. is a soluble case ...) — ANN has a (good) approx. soln.

2) TRY to state the problem very clearly & show what the difficulties are.

3) Write review of impt. results (N.B. recent work on say a little further back on "SMA Problems" (in TM/SM) — this should furnish some approx'ly solvd probs.

4) Consider the "Honest" evalns: (a) If on a priori choses an AA, its observed "value" is an unbiasd estimate of its future yield if the sep. is "stationary".

(b) Similar results obtain w. non-stat. seqns. if one considers "ensemble averages". in (a) if one choses AA's A_i w. apprp p_i resp., & observed yields, Y_i, then $\Sigma p_i Y_i$ is an unbiasd estimate of "$\Sigma p_i AA_i$" — This is "Cover's Uniformity."

5) Any (Do I have) Reasonable Approxns. for the final TM version of the problem?

6) A possl. idea! Using past corpus of problems: compute pd. of future problems. Then evaluate (probabilistically) the yields of the various AA's on these future problems.

7) Modifn. of J's method! Divide past corpus into (sequences (?)) parts [redacted] of same Time-size. Try [redacted] a sequence of AA's on them

| letter to T Etter 12 2 96
| compute prob'ty that yield in future will be y.

A somewhat different tack: from ~ 188.27 – .40 casts ⊙ on Utility of T's goal of max ΔR during life of TM. So in general I think that for a "Scientist's helper," I'll simply want to control externally, & audit, how much time T.M. spends in S.i. — BUT NOTE (10)!

But even if TM knows it will live only time T, it must solve as many Env. probs as poss'l. during T; it will find it best to first do some S.i., then work on E. problems. In general, I don't I wish TM to work on a "large" problem, it will by itself, allocate certain time chunks to S.i., because that's E. Protect way to solve "Big problems": One first improves one's education &/o HW, or tables or "aids to computation" — or whatever.

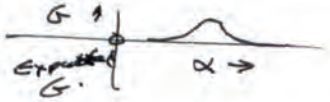
10 → Anyway! If I give TM lower & lower probs, it should automatically [illegible] how much time to spend in "S.i.":

But when TM is young, I do expect to have TM solve specific probs. Then spend time on "S.i." — i.e. putting E. P.D. in better shape. — Perhaps this "S.i." S should be regarded as part of the next problem! — So T.M. would spend no time on "S.i.".

On the other hand, when TM is "expressing the p.d." the manner in which it does this is indep. of the what the next problem is.

.17
18 On the Q of max pc. G not being best thing to do, e.g. ANN (use of Broad Max's best) Also in ALP we use the effective width of a peak (= pc of peak.)

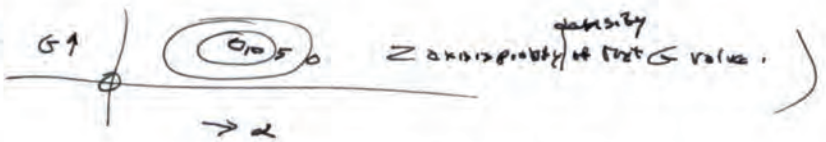
A In the General problem of Max of expected future AA: Say one's subjective approp of G was



(α is a param (usually vector) that partially describes the A A

OR

Possibly:



.28 Anyway, if we want to do the AA w. pc. G = Max, we can ↑ the pc of a particular AA down by ↑ its resoln (≡ no. of bits). So for a subjective width of 2 bits in α, we would choose a lower resoln. of α. This tends to place it not at the peak, but since the width is 2 bits, it tends not to make much difference.

.31 There is, in ALP an exact way to do this exist: Maybe reading Chaitin, Wallace would help figure out an exact analysis. Anyway, the result is that we want to select an "α" → G × pc at peak × width of peak to be max. For the p.D. in which we have no real peak like an integrable pole — say at the origin; it's not clear as to what to do.

.33
.34

Or, if we have a discrete digital distribn,

[A commonly usable "ruff & dirty" Soln]

If there are >1 peak: .33 – .34 is usable.
.33 – .34 " "

.29 – .31 may be usable for digital distribns as well. → 193.08

[illegible]

A not bad approach is 189.10-.13 : (noting that 189.13 is a normal Future AA optzn problem (189.31))

Also note 192.10 : that giving TM progressively larger problems ("Larger means solns cost more cc) results in it's automatically doing Self-improvement. It is not necy that TM chose these problems from an unordered pool..... (The eventually, I.M. should be able to work that problem as well!).

.08 : 192.40 It may be that the reason 192.28-.31 works is that AA Dcms that are algorithmically close tend to have G's that are close for corpi that are algorithmically close

. for digital stuff, .08 may not be so reasonable. But can choose a corpus to change its G enormously!

Hvr, if we assume the spaces of α's of G are continuous, that rests some reasonableness, & some natural Metrics.

• .13 In SM strategies, the G can be a discontinuous function of the continuous params defining the strategy, but we expect that for very long averaging times, the G function of α will have only small discontinuities. This is opposed to Lovell's Gone which is very continuous - smooth.

So the SM problem of .13 w. short corpus, is discontinuous & important, & is more like the Digital, (discrete) case of AA optzn.

Another simple case of a very smooth, continuous G is the Maxm problem (& linear regression or curve fitting) In this case the peak α is the best predictor — but estimating it the "width" of the resultant predn. is more complex. (The more or less exactly computable).

→ Maxm assumes that all of the ragys in the corpus have been dealt w. by the predn. method.

.22 Perhaps we can take any AA or any "pure AA" & find a norm which it characterizes the ensemble of corpi that it "optimizes wrt." ?☺?

In norm, the Gone at the peak & the way the Gone drops as we move away from the peak, gives us the true future σ^2 & perhaps enables us to completely characterize the ensemble. See 191.01-.06! From the Distribn. of G among the various α's tried (& knowledge of AA structure), one can get the ensemble of the corpus.

Each α_i, G_i pair defines a set of corpi that would give that G_i on that α_i. The continuous set of $[\alpha_i, G_i]$, that one has then gives the intersection of ensembles $_i$ (defined by α_i, G_i)] which is a much narrower ensemble.

Hvr. If α_0 G_0 are the peak, then α_0's AA (≡ AA_0)

.32 Applied to this intersection ensemble we will give just exactly G_0 — "$\sigma^2 = 0$".

My Post on this has been a bit off: In norm, we have the ms error for each $\vec{\alpha}$. α_e^2 is ms error for $\vec{\alpha}_0$" — which is minimal. The discussion of .22-.32 still holds, hvr.

Looking at Maxm: One way to do the analysis is to look at $\vec{\alpha}$ (≡ "the coefs + α^2"). The coefs part is "uncertain" & the distribn of the coefs w. error tells how much uncertainty it is. In the case of Maxm, this dF. in probability & one can be directly translated into error in the set of coefs. (I know did some work on this while last 6 mo. (or so or 3 mo.)). "the particular SM problem".

It could be in SM files ↳ It may be in "Nite Notes".

Someway to map ΔG into changes in probty? (TM 33.22 ff is on this topic. ← It is a discn. of a rather dirty soln. to ABE (AA evoln) = SMA problem. Interesting that! It uses history of G as a Time series & extrapolates its future & MSG via linear or N.L. (Markov eq) methods.
It does not use t. structure of t. AA to devise predn. schemes, hvr. ——

Non-t.less, this idea does give one kind of Criticism of J's "LHL" — he uses simply t. mean rate of R of $\overline{R}$ as a Gen — essentially as a predictor of future rate of R of Gen. This is not a v.g. way to do predn!

Note: that a G has been "linearized" is not nearly related to its being closely related to probability.

Assoc. w. each value of α_i, G_i, is a set of corpi that could give that pair. t. true corpus is in this set; it has a pc within that set. ——
So, t. α_i, G_i pair's pc. mult by that pc is t. pc of a code for t. corpus assoc. w. that (α_i, G_i).

To do this, one must have a "default code" = default p.d. for t. corpus: For SM, this is easy to get: it's t. "Random Walk P.D." The G peak could be at or near the pc. peak: This is because for most default d.f.'s hy G's are rare & so t. measure of t. corpi [illegible] w. hy G is small, so it doesn't take many bits to tell, within that sub-corpus — which one t. true corpus is.

T. former coding is Cover's "Extension probty." It works o.k. in theory, but much harder to implement than straight ALP — unless t. individual predictions are sequentially independent — otherwise, its a lot of trouble to consider all possl. finite continuations of a corpus. We mite have such a situation in ① SM, where a bet can end and be followed by a new situation indep of t. last bet. (This is not, alas, true of "Reversing" strategies). ② A TM that works discretely bounded (in time) problems. — So what problem comes next is not much coupled to what came previously (Unless, of course, we give TM a carefully constructed trg seq.)

.31 Good! Departing a bit from "Extension" probability: Say α is fixed, but we get periodic updates of G as time goes on: This is appropriate to most SM strategies & Many forms of TM w. Reinforcement.
So t. partial code of t. corpus is a fixed α & a sequence of G values.
For SM, given α, one could do Mt. Carlo runs to get D.F. of various G (Yields) —
Here t. default d.f. is t. Random walk.

.31 is much like doing induction on an underdard set (If t. individual trades are statistically indep. For induction on unordered sets, I think Extension proby ≡ ALP.)

.01 Again: Re: [redacted] 1943) ff: It would be v.g. to analyse Cover's U. part'f. I think his trades are meant to be sequentially indep.

One easy way to deal w. this: each $\vec{b}$ value is equiv (in default) to a single stock w. a certain Varc. (default means always so)

So let's study the single stock w. mean ϕ & Varc 1 for the time interval of interest. We have N observations of its behavior for (day's trading, say,

— Maybe use a 1 Wk interval, so daily data does exist, but we don't [redacted] have access to it.

In this coding Market using a "strategy":

We can start with (a) apriod of yield of strat. as a function of $\vec{\alpha}$.

(b) the "Default" apriod of the corpus.

I think (a) need a f.d. of yields for each value of $\vec{\alpha}$. — [redacted] a rather complex object — "User" May not have this Available!

New track: Say one has a (space) / set of cards: def'd by $\vec{\alpha}$. So the dcrn. cost of each is the same. Each $\vec{\alpha}$ point, however, has an assoc. subspace of corpus space. If $p(\vec{\alpha})$ is the [redacted]pc of that subspace, then the coding cost (p. c.) assoc. w. the point $\vec{\alpha}$ is $\propto \frac{1}{p(\vec{\alpha})}$ (This is quite familiar: I obtained this result

.20 previous time around).

Note: I can use any apriod that I like for the ensemble here

However, what I want is some thing like linear regression, so I can use ZMEKM on the corpus.

In linear regression, the method of predn. is optimum, if the corpus ensembles of a certain type.

For an arby strategy: say we are able to try it repeatedly & the trys are statistically indep. — Then we will get a Distribn function of yield for that strategy, & we can readily extrapolate into the future. [This approach may be the same as the SMA, AAE prob "John" Reviewed in ~ 87.01 ff, 89.08 ff] — Perhaps my Complaint about that "Soln" was that it didn't use the structural info into/of the strategy of "AA" to get the soln. — so it was leaving out imps. info .... [The it mite be o.n. as a "Ruff & Dirty" soln]

[The most recent approach seems to use that info but I haven't been getting a [redacted] result from it that is of much value!

In coding via $\vec{\alpha}_i, Y_i$: While the pc of the $\vec{\alpha}$ values may all be about the same (uniform), the apriod of the Y_i values is probly not at all Uniform. We need an apriod for Y_i before we can get a proper code for the corpus this way. The expression of .20 $\frac{1}{p(\vec{\alpha})}$ has to be mult. by the aprio of Y. We could use the Universal default distribn of Rissanen ($\propto 2^{-\log_2^* Y}$) but this would not be really solving the problem correctly. (or simply use the divergent $\frac{1}{Y}$ or: $\frac{1}{Y}$ mult by a factor that depends on $\sum \frac{1}{Y}$ .... so it normes properly).

Spec 19/6.22

83097 TM LHL:

Re: T. Time horizon problem & T. top Goal of Max ΔR in a life of ∞ length unknown in advance!

For my own purpose: of constructing t. "Scientist's helper": A TM that was able to track a Tng. Sq. of my own devising, of problems of gradually ↑ length (≡ difficulty), would be adequate. T: problem pool " form of Goal / might be nice, but it is unnecy. Using t. simpler form, I think I could get TM up to any desired degree of skill.

So all I need is a TM that will try to solve whatever (Inv. problems I give it w. min cc; & solve any OZ problem I give it w. max G in Given time, or perhaps an solver t "Anytime" form.

T. OZ soln. form or request form, is not clear in my Mind — possibly because there are so many forms of OZ probs — many not so probs (to my knowledge) by Srch.

.18 I still have to solve t. AA Evaln. problem! It seems to be necy for optimum "updating". — even for INV problems alone! i.e. T.M. has to modify t. P.D. so that cc of solns of probs in future will tend to to be Minmal. .... Which is a problem in AA often.

.22 : 193.40

Back to SMA/AAE:

Correspondences betw AA coding & Maxm:

AA: $\vec{\alpha}$ ← dcrn of AA & ■ ↑ give part of code for corpus.

Maxm: $\vec{\alpha}$ ← set of cells & σ " " " " "

In Maxm, $\vec{\alpha}$ gives t. cells; then ■ to ■ dcrb t. corpus ■ completely, we need to give N error readings of ms value σ^2. (for a discrete corpus).

An example: Maxm: $\vec{\alpha} = \vec{0}$, so we just have ensemble w. $\sigma^2 = 6^2$ for each discrete pt (say each day). If a corpus has vare = σ^2, then it is in t. most (dense numerous) part of t. ensemble: vare = 0 has only 1 member, vare = $100\sigma^2$ also has very few members. — So if we dcrb a corpus as having vare = $.0016^2$, this would be a very short dcrn.: ■ if all vare values were of = ■ aprip

If vare of s^2 has p.d. ■ of its density in t. ensemble, then this method of coding is of little value: All vare value got same wt.!

Back to "AA Coding": We observe our/particular corpus has max Y when $\{\alpha = \alpha_0,\ Y = Y_0\}$.
So perhaps α_0, Y_0 is "a" t. most likely "generator" of t. corpus, (selecting from t. space of possl. generators, $\vec{\alpha}, Y$.)

Seems to be a difnce here, btw usual Maxm coding & t. AA Coding I'm considering: For Maxm & linear regressn.: when we specify $\vec{\alpha}_0, \sigma_0^2$, t. ensemble is not corpi of $\text{varc} = \sigma_0^2$ only — tho corpi of $\text{varc} = \sigma_0^2$ are in t. densest region of t. Ensemble.

(actually, varc of t. ensemble is not σ_0^2; but {t. varc. of t. error, using $\vec{\alpha}_0$ as a predictor} is σ_0^2.)

But go back to t. simple case $\vec{\alpha} = \vec{0}$, $\text{varc} = \sigma_0^2$: Discrete time.
Consider t 2 ensembles of length N: ① $\text{varc} = \sigma_0^2$ ~~[illegible]~~ exactly for ~~entire corpus~~ each member of ensemble. ② Mean $\text{varc} = \sigma^2$ over all members of Ensemble.

Re: "All practical applicns of Proby involve proby ratios" — Counterexample: "proby it will rain > .1" tomorrow

---

.17 For example of AA: $Y = \bar{x}$. $Y = 0$ is most likely. for simple 1 period,
.18 of $\text{varc} = 1$, proby density for a gn. y is $\left(e^{-y^2/2}\right)^{-1} = e^{y^2/2}$.

Th "BDS statistic" is another AA of interest for predn. {see B, H, L Book}
In fact, I did do much analysis of this statistic w. t. idea of using it for predn., but I wasn't actually successful. (tho I think I ended up w. some nice conds. that I didn't, hvr., try out). My approach to this particular problem was rather strong! I.E. If BDS assigned $pc = \epsilon$ to t. corpus wrt. t. default ensemble than I should be able to get a yield of $\frac{1}{\epsilon}$ from t. system

.26 Hvr. .17 is t. easiest to analyse! If we used $Y = \sum x^2$ we would get a chi sq. d.f. for proby^{-1}. If $Y = \bar{x}$ we get probn = .18.
— But $e^{-\frac{y^2}{2}}$ is unnormalizable as is.

I'm Confused!
.30

SN (INV OZ) problems: T. type I've been considering are ones in which th. entire corpus of relevant data is accessable, but there are too many cand. solns & some take too long to test.

I have not considerd cases in which t. corpus was very large, so one usually samples only part of it in attempts to solve t. problem. This is certainly not an unusual situation! It occurs in SM, Data Mining, Input from R.W. ....
What to notice in t. Envt..... Info Retrieval

Also even if t. corpus is not very large (in a superficial sense) if t. available time is small, we will in have an essentially "very large" corpus — so that sampling is to be considerd.

9.24.97 TM LHL!

197½

01 E Earlier refs on "Jargon": 183.13 is one "recent" sort of letter to J.

Seems to me, Just More recently, I had idea for letter to J that was more positive in its introdn: An explanin. of just why his (non-el.) approaches v.g. & what kinds of duffers it successfully avoided!

207.17 J's good
196.18, .21
196.01 *
192.17
188.01, 189.01 Time Horizon

B.g ① Deciding on time horizon ② How to evaluate ■ Utilities. ③ How to divide prob. into sub-probs ④ How to Learn.

That in "long run" his method is probly best way. —— but "long run" is very long.

What I'm mainly int. in is to "Scientist's assistant" in Min Time.

Some points on t. AAE/SMA problems

1) If we assume data is stationary, t. Y assoc w. $\vec{\alpha}$ for T=∞ is meaningful.

2) T. idea of max flatness, t. Hessian & Maxm can be used to get t. proper vue of Y. — As a kind of n.l. curve fitting problem.

3) Dorb. problem to J. : Menton his work on max-flat pts & Hessians is relevant.

4) for ② & See what work I've done on this around 31.01 ff

A goal might be to try to extend t. "max flat" to discrete distributions.

a/o partly discrete, partly continuous.

An assidon on AAE:

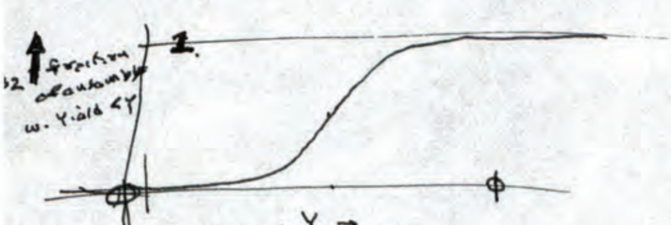
10 If we had ∞ resources, it would seem that a formal soln. is possi. ① First,

.11 Using ALP, obtain p.d. for t. future of t. sequence. ② Then t. problem of finding an AA w. greatest "Expected Yield" in t. future, is a well define OZ problem, even w. finite CB.

In R.W., hvr. CB for part 1 (⑩) < ∞, so we may consider that a sub-problem would be "How to divide cc betw ① & ②? Maybe do them in ‖ (time share), or use 50% soln? Can I show 50% soln is "not bad"?

Re: ⑩ in SM envt. predicting t. future of t. corpus is ≅ t. main problem. If we could do that, ② ⑪ would be unecy. T. idea in much of my recent work on this was that t. set of [$\vec{\alpha}_i$, Y_i] values that I've obtained is in a very useful sense, a characterzn of t. ensemble, & could be used to obtain its p.d. (or whatever we need to solve t. problem). It would seem that this info plus t. apriori (t. "default" d.f.) should be enough in order to get what I need.

Superficially, $\vec{\alpha}$, Y seems like a reasonable coding method — tho 197.26-30 & t. discussn. leading up to it seem to indicate lots of trouble! "#¢#.###ω"



pcost = Volume in ensemble space: This pcost is indep of t. shape of t. volume.

So to code a small region of ens. space (which is t. only way to code a corpus, since any particular corpus is always a region (usually small) in ens. space.

To code such a small region, first find the smallest Y interval that encompases t. region. Then say P_y is t. pc of that region, & P_c is t. pc of t. corpus wrt. that ensemble.

Then our "code" assigns $pc = \frac{P_c}{P_y}$ mult by t. pc of the Y specification: which is partly proportional to t. width of t. Y interval that contains t. "corpus".

[we mite as well regard t. slope of .32 as t. prob density for a "single corpus" of standard size ≡ "resolution"]

.01 So 198_32 begins to look like "if an unlikely thing occures it is of interest/value".

Perhaps another way of looking at this: If the set of Y's from a sequence of subcorpi cluster about a region that the default d.f. gives low pc to, then it is clear that we would do better than the default d.f., by coding the set of Y's in a d.f. not way: AE.

A simple way would be a norml d.f. about the mean of the Y's. Better forms for the d.f. might be obtained by examining $\vec{\alpha}$ & the structure of the AA. → 200.06

.07 → ! What about this?: We have the [α_i, Y_i] set of data: To get the P.d. of a particular α_j for the next trial. This is a standard ALP problem w. known soln. (Probistic) Operator induction. One way is Nonlinear Mexm.

An "Objection": Suppose one only has one α value & many Y values? Then the problem is to characterize the D.f. of Y from the set of Y's.

Usually (almost always) one has Y's for many α values: This is nice so once can find the α region for max Y.

① But how can we apply .07 to, say Covary Universal portfolio? →

.15 SM Somehow (until recently) (.07) I felt That the AAE problem was significantly diffrnt. from the normal sequential prodn. problem in ALP. Just why was this? I guess I felt predicting the SM was ALP; but getting P.d. of future value of a AA was different. As I see it now, AAE is just another time series to predict. It has, hvr, in SM, a certain underlying structure, that one should, presumably, take advantage of — i.e. the default "Random Walk" SM model. — See ~ recent mem on this ~ 6 mo. ago

.24 → ② What about problem of having many possl. starts to choose from & picking the Apparently "Best" one — (for SOY problem)? Well, using ALP one obtains the joint P.d. Yield function vector of all the starts being considered. Then we pick the "Best one" (for whatever Goal we want), Then use SOY (if we like) to find out what our expected yield is (This last step may not be needed hvr, for many applics — having the "Best" start (or AA) may be all that we need.

.37 Hvr, 171.01, .20 suggests that picking the apparently best AA is not always the Best thing to do!

Also Note: 188.01 -.26. → One reason we may want to know the expected yield of the "Best" AA (viz the SOY correction) is that the yield may be <0: a 171.01 suggests that if the Zero yield starting AA (w. Zeros) is taken as one of the 5 to block A this considered, we will not automatically choose it using the SOY correction routine I've devised.

So this "Soy Correction" is still Not really Solved — i.e. problem of Selecting the Best AA is not solved.

9.1.97 TM LHL

(171.40) .01: RE: 171.01,.20: • T. technique of 171.01ff first chosing t. apparently "Best", then subtracting out "Soy", Then comparing w. each of t. possl. strategys..... This ≥ ≧ way of picking t. Best of a set of strats., but I'm not sure as to how good it is! — T. counterexample worns easy because one strategy has σ=0. How do we do comparisons w. single stratys. where there σ's are > 0?

Consider 171.24 (a less el. way to proceed). Is this any better? It would Seem so.

06: 199.00 So: 199.01–.06 suggests That my previous ideas about using yield of a strategy as a coding method ← maybe correct but Not Far From TRIVIAL!

10: 188.19 On t. "width of a peak" consideration. I think if one uses ALP properly, one would automatically Take peak width into consideration.

On to SMA (Stk. Mkt Advisors) problem. (One need not us SM advisors: Funds, Stocks, indices futures, options or any Financial instrument having sequencial evaln., could be used).

Even if one could deal w. t. "Dropout" problem — There is t. Soy Problem — I don't know if I had considered This in my most recent foray (apparently fairly successful as I remember). — So one mite very well pick t. "Best" SMA, but his yield could be quite small!

9.030

.20 The SOY/SMA/AAE problems are all Decision Theory problems

They involve ratios of pc's (∴ Theoretically, somewhat undefined)

.22 → J's formulation of t. LHL problem can be viewd as a non-el approach to "decision Making": w.o. explicitely using Utilities.

This can be V. G. because I know of no v.g. way to get Utilities in General.

Utilities are invented as possl. "Sub-goals" — ~2000: I decided That "Utilities" are params in a model That derbs a human as a goal optimizer. Nothing More! One can have more general (non-oz) models of Humans.

So Re: .20 In Cover's U. portf! which $\vec{b}$ to use?

If a simple Gaussn w. Mean ≠ 0 is usd to code each of the space of $\vec{b}$ strats, Then we should chose $\vec{b}^*$.

In the ordinary Soy problem, If we have 10 strats w. true means φ & all Gaussn, σ=1, Then from a sample of say 20 data pts, we will pick one of These, rather than a strat w. σ=0, μ = .01.

In my latest "Soln" of t. SOY problem I think I took a Meta ensamble.

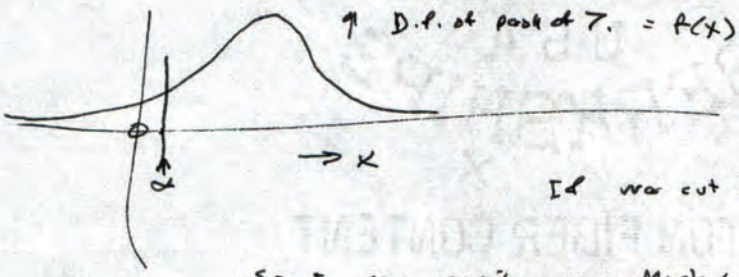
9.6.97 TM ~~Eth~~ / SOY

.01: [171.02] On t. counter example to my "Soln" of t. Soy problem: 171.01

I really should work it out more exactly!

Say I have $\beta > 7$ / indiv normally distributed r vars w. means $=0$. The peak of t. $\beta > 7$ has a certain d.f. . Using Maple, I was able to calculate the mean & var. of this distribn. If, to the ≥ 7 vars, we add an α^{th} of mean α & $\sigma = 0$, then t. mean of t. d.f. will $\uparrow$, since we only integrate its first moment from α on up.

.08 $\uparrow$ D.f. of peak of 7. $= f(x)$



mean of $f(x) \equiv$

$$\mu_{-\infty} \equiv \int_{-\infty}^{+\infty} x f(x) \Big/ \int_{-\infty}^{+\infty} f(x)$$

.10 If we cut off d.f. at α we get $\mu_\alpha = \int_\alpha^\infty x f(x) \Big/ \int_\alpha^\infty f(x)$.

So, I can easily use Maple to calculate $\mu_{-\infty}$ & μ_α &

$\mu_\alpha - \mu_{-\infty}$ = always $> \alpha$?

t. ~~Q~~ is: it's ~~always~~ ! Intuitively, I felt it ~~was~~ could be $< \alpha$ for α slightly > 0.

Disturbing! $\mu_0 > \mu_{-\infty}$ certainly!

No! Soon .35

Actually t. defns .08 & R are wrong: when we put a stop at α we get a distrib d.f. than just cut off. We get a δ funct of amplitude $\int_{-\infty}^{\alpha} f(x)$ at α:

So $\mu_\alpha = \alpha \cdot \int_{-\infty}^{\alpha} f(x)\,dx + \int_\alpha^\infty x f(x)\,dx$

$\overset{*}{=} \alpha \int_{-\infty}^{\alpha} f(x)\,dx - \int_{-\infty}^{\alpha} x f(x) + \int_{-\infty}^{+\infty} x f(x)\,dx$

.22 $= \mu_{-\infty} - \int_{-\infty}^{\alpha} (x-\alpha) f(x)$ which gives $\mu_\alpha < \mu_{-\infty}$! : unreasonable!

No! $\int_{-\infty}^{\alpha} (x-\alpha) f(x)$ is usually < 0 because x is mainly < 0 & bcaw. $-\infty$ & α.

Say $\alpha > 0$: $\int_{-\infty}^{\alpha} (x-\alpha) f(x) = \int_{-\infty}^{0} (x-\alpha) f(x) + \int_0^\alpha (x-\alpha) f(x)$

.22 $\rightarrow \mu_\alpha^* = \alpha \int_{-\infty}^{0} f(x) + \alpha \int_0^\alpha f(x) \boxed{- \int_{-\infty}^{0} x f(x) - \int_0^\alpha x f(x) + \int_{-\infty}^{+\infty} x f(x)}$

$= \int_0^\infty x f(x)\,dx$.

$\therefore \alpha \int_{-\infty}^{0} f(x) - \int_0^\alpha (x-\alpha) f(x) + \int_0^\infty x f(x)\,dx$

.35 I'm still not analysing this rite. The correction constant = mean of (observed variable − mean of that variable). Whereas variable $> \alpha$, mean of variable $= \phi$, so we take mean/value of all observations $> \alpha$. $= \int_\alpha^\infty x f(x)\,dx$

To this we add $\int_{-\infty}^{\alpha} f(x)$ — which is fraction of time that observation is $< \alpha$ &

so $\alpha -$ (mean of the variable which is $< \alpha$) $= \phi$ so we add zero to this

or: $\int_{\alpha}^{\infty} x\,f(x)\,dx$ + [illegible] $\int_{-\infty}^{\alpha} (\alpha-\alpha)\cdot f(x)\,dx$.

.02 assuming $\int_{-\infty}^{+\infty} f(x) \equiv 1$. So perhaps 201.10 is right! $\mu_\alpha = \int_{\alpha}^{\infty} x f(x)dx \Big/ \int_{-\infty}^{+\infty} f(x)dx$.

So: we do this sampling & we get a mean of [illegible] $= \alpha \int_{-\infty}^{\alpha} f(x)dx + \int_{\alpha}^{+\infty} x f(x)dx$

subtracting out μ_α of .02 gives only $\alpha \cdot \int_{-\infty}^{\alpha} f(x)dx$. — it should give just α.

Perhaps that's the wrong idea, anyway! observed $x - \mu_\alpha$ is an unbiased estimate of yield. The trouble is that choosing the variable that is apparently largest is not the best way to make a choice here:

But I am beginning to get confused! Unclear as to exactly what the problem is!

.16 → [It always was a rather confusing problem] → see .23 !

.17 Suppose one had a bunch of (k) strata of means ϕ's & $\sigma^2 = 1$. (If the bunch is large ($k >> 1$)

.18 we get for a certain finite no. of time points (N time pts) a d.f. of means w. σ of means $= \frac{1}{\sqrt{N}}$.

Say one doesn't know the d.f. of the strata:
One only knows the empirical data for N pts on each of k strata.

However, [illegible] is one test of the hypothesis (made at .17–.18, then it would fit the data rather well (large PC.).

.23 AH! That's it! We have to make a model of the entire data set! Just as we did in the original SMA problem (worrying about "dropouts" in that case, but not in the present case). → Also see 206.37

In the case of 171.01, the best model will be in 2 parts: One part generates the stochastic sequences about x=0; The other part generates the $\sigma=0$ data at x=1. If there is not enough data to legitimately divide it up into 2 parts like this, then we just use whatever best model we can find — or, more exactly, to/with sum of all models ($\equiv$ ALP).

In [illegible] in 171.01 ff we have several time series' but this does not add any additional difficulty to the coding. We just have a vector to be predicted each day.

Also, in 171.01 the first part contains several T.s's: If they are different enough, we can assign distinct params to each. If not, we have to regard them as having same μ & σ (or same μ, several σ's; or several μ_i, one σ, ... or subset of some μ_i, σ_i, etc ... whatever ways increase pc.)

I'm still not sure that 202.23 solves the SOY problem. If one has several strats. w. highly overlapping p.d's: They can have different legit. μ_i's — but still one has SOY. The complaint that "at least we know the best decision -- the one we don't exactly know its yield" — is incorrect. We have to be able to know a quantitative yield of the "best", since it must be capable of being compared to a strat. of yield X i.e. $\sigma = 0$.

Around 171.01, my mind was a bit vague on how to get the joint p.d. for all of the strats. — Hvr. ALP does give it.

But, Given the ("correct") joint p.d. of the set of strats: There does seem to be a considerable opportunity for SOY!

Say we just have 2 strats. For tomorrow, we have a joint p.d. for yields of the 2 strats. (or for the next 100 days, say). This would seem to give SOY. {Only 1 strat} gives no SOY. [redacted] Yield [redacted]

Maybe NOT! : The thing that gave SOY, was uncertainty in the [redacted] yield mean yield of each strat. ALP gives the D.F. for tomorrow (or next 100 days) explicitely w.o. necessarily considering "mean yield" of any strat. "The d.f. of the mean" is an al. concept. It need not be meaningful. 12:12p 13 14

However, μ can be used (& commonly is) in ALP codings.

An impt. Quality of ALP predns: They include uncertainty of the μ as well as uncertainty of predn wrt a known μ. Hvr., this doesn't seem to get rid of SOY problem!

If the μ's of 2 strats are within that of of each other, from the hypoth. that they are different strats, is not so strong. This overlapping of D.F.'s for the μ's is the main situation I'm interested in.

On the other hand, the 2 strats could be clearly different, yet have μ's that are ~~close~~ within < Phone (& common) of each other. One still wants to do well by pooling their data until SSZ is large enuf so that the σ's of their μ's is $\ll$ the diff between their μ's

Could we ~~assign them a (protocol) column~~ pool their μ data "descriptors" but allow them different σ's for predn.?

The main Q is: How do I get "Unbiased Estimate" of Yield?

An argument that the bias will always be "small": 2 cases: (They can't (usually) be much closer than their & common σ: otherwise they are prob the same & one should pool their data.

1) μ's are all huddled together < σ apart (& common σ)!

2) μ's are separated by > σ's of top μ_i.

In case 1), one gives most wt. to pooled model (only 1 M) — which has no SOY bias. Other models have more SOY, but less wt.

In case 2) there is little SOY bias.

Q: are these always the main cases?

Q: Even if the bias is small, it is still Bias: & I want to get rid of it!

Also: The fact that I don't know how to get rid of it makes it likely that I don't really understand it!

I should be able to estimate its size via the Ensemble Average & ∴ get rid of it — i.e. subtract it out.

If so, then I'd have to see if the Criticism of 171.01 is valid.

MEXM seems to automatically deal w. the SOY problem by summing over all possibs. (Remember, its the σ^2 in Mexm / that corresponds to Y). — i.e. $\sigma^2_{\text{in Mexm}} \to \frac{N+K}{N-K}\sigma^2_{\text{obs.}}$

Mexm solves a "purely ALP problem": But SMA is also a "purely ALP problem".

In Mexm, the α w. best Y ($\equiv \sigma^2$) is the one w. apparently best Y, but the Y is worse than apparent, by the factor $\frac{N+K}{N-K}$.

.15 Consider 3 problems: (1) linear Mexm (2) Covar's Unrestr. Portf. (3) SMA w. discrete
.16 strategies.

A better example of Mexm than standard scalar predictor: Use Multilinear regression: I think this is very close to linear Curve fitting. The best example would be prediction of a vector.

.21 Say Multilinear regression, we want to predict a vector: the values of the K different TS's tomorrow. We get a linear predn. for the center of the d.f. is a (perhaps Multivariate) d.f. for the distribution funct. I want to bet on the seq. w. highest expected value tomorrow. The Quad. form giving the M.V. DF is constant, if the linear eqns are "stationary". From this Quad form, we can compute the SOY & its variance. This will give us the average bias between the predicted peak, & the actual value the next day. For $k=1$ this bias $=0$.

On second Read: while the Quad form may be constant, the SOY will not be. Either The SOY depends on both the predicted vector $\vec{x}$ (the center of the D.F.) and the Quadratic form that gives the "uncertainty" in $\vec{x}$.

.31

It is perhaps in .21 that the confusion arrises. In Mexm, if we considered σ^2 for next predn. to be Y, then Mexm tells us how many // hyperplanes $\wedge \sigma^2$ beyond that of the pack. ($\equiv$ the $\frac{N-K}{N+K}$ factor)

In .21–.31 the Y is different! it's not σ^2 or the cov. matrix at all!

In .21–.31, the centers of the d.f.'s for each scalar predn are unbiased. Also, the variances are unbiased. However, when one picks the seq. w. the largest mean value, the expected yield will not be the expected mean value ($\equiv$ expected value, if one chose that seq. every time).

Check on whether the objection of 171.01 is correct, now.

One big reason I have to know expected yield of peak, is that I want to be adding new securs., & I want to know if this will ↑ mean yield, or simply add noise & ↑ apparent yield.

A big Q: Does adding new strats never ↓ yield? In MaxM, adding another coeff can ↑ σ^2, but this is not exactly the same thing.

Note that in MaxM one can use either σ^2 or $\bar{X}$ as the output $\bar{Y}$. (a- Rm).

91097 In MaxM, adding a new coef, or new time series (indep var) will always ↓ observed σ^2 — Hvr. $\frac{N+K}{N-K}$ can ↑. In ALP, [redacted] we can, in a simple case add a new trend. model with its pc (a priori) and its σ^2. The σ^2 of the result is ≈

$$\approx \frac{N_{old}}{\sigma^2_{old}} + \frac{N_{new}}{\sigma^2_{new}}$$

/ Nold & Nnew are ∝ SSZ corresponding

to the original & newly added Models. They are ∝ wts. which are ∝ a priori:

$\frac{1}{\sigma^2}$ is ∝ wt.

Say original model has predn of mean μ_0; Varc = σ_0^2; pc = p_0. ← ≡ wt$_0$

We introduce auxiliary model μ_1, σ_1^2, p_1.

Is the forgg. the same as the following: We make n_0 observations of a quantity w. Varc = σ^2. & mean μ_0. We then make n_1 observations w. mean μ_1 & varc σ_1^2.

We could express this as a single set of ~~these~~ n_0+n_1 observations: The

Final ~~Var~~ Varc. is $\left(n_0 \frac{\sigma_0^2}{n_0} + n_1 \frac{\sigma_1^2}{n_1}\right) / (n_0+n_1) =$ [redacted] $\frac{\sigma_0^2+\sigma_1^2}{n_0+n_1}$

No! Say σ_0 is the mean error of first set of n_0 observations, μ_0

σ_1 " " " " second " " n_1 " μ_1

.24 I think mean is $\left(\frac{\mu_0 \cdot n_0}{\sigma_0^2} + \frac{\mu_1 \cdot n_1}{\sigma_1^2}\right) \Big/ \left(\frac{n_0}{\sigma_0^2} + \frac{n_1}{\sigma_1^2}\right) = \mu$ } Probly Wrong!

.25 The varc. is $\frac{n_0}{n_0+n_1}(\sigma_0^2 + (\mu_0-\mu)^2) + \frac{n_1}{n_0+n_1}(\sigma_1^2 + (\mu_1-\mu)^2)$

See .35 and .37

There is probly a very simple expression for the varc.

A way to analyse this: pick a μ s.t. .25 ← varc. is minimal (least sq. err)

The result will probly be .24.

$$\frac{1}{2}\frac{\partial .25}{\partial \mu} = \frac{1}{n_0+n_1}\left(\left(n_0(\mu_0-\mu)\right) + \left(n_1(\mu_1-\mu)\right)\right) = 0$$

$$n_0\mu_0 - n_0\mu + n_1\mu_1 - n_1\mu = 0 \qquad n_0\mu_0 + n_1\mu_1 = (n_0+n_1)\mu$$

.35 $\mu = \frac{n_0\mu_0 + n_1\mu_1}{n_0+n_1}$ (!). Indep of σ^2's! (Unlikely!)

Well! .25 is probly wrong. But .37 seems to say it's correct!

.37 Say $X_i = \mu + \Delta_i$

$\sum \Delta_i^2 = n\sigma^2$ $(i=1..n)$

$\sum \Delta_i = 0$.

$\sum (\Delta_i + \Delta)^2 = \sum (\Delta_i^2 + 2\Delta_i\Delta + \Delta^2) = n\sigma^2 + n\Delta^2$ so changing varc is into a minimum at $\Delta = 0$

+34
12
98.70 +.05
20. −.02
21.50 +.05
51.03 +.2

Kernel32.dll
version
2/2/96
header

kernel32.dll
402M
11/7/95
c:\Program files\
Alaska Internet
\ Alaska

.01 : 205.25 is probly correct. $\mu - \mu_0 = (\text{via } 205.35) = \frac{n_0\mu_0 + n_1\mu_1}{n_0+n_1} - \frac{\mu_0 n_0 + \mu_0 n_1}{n_0+n_1}$

$= \frac{n_1(\mu_1-\mu_0)}{n_0+n_1}$

$n_0(\mu-\mu_0)^2 = \frac{n_1 n_0 n_1^2 (\mu_1-\mu_0)^2}{(n_0+n_1)^2}$ $\quad n_1(\mu-\mu_1)^2 = \frac{n_0 n_0 n_1 (\mu_1-\mu_0)^2}{(n_0+n_1)^2}$

so 205.25 $= \left(n_0\sigma_0^2 + n_1\sigma_1^2 + \frac{n_0 n_1}{(n_0+n_1)} \cdot \frac{(\mu_1-\mu_0)^2}{(n_0+n_1)} \right) / (n_0+n_1)$ $\quad \frac{n_0 n_1}{n_0+n_1} = \frac{}{\frac{1}{n_0}+\frac{1}{n_1}}$

.08 $= \frac{n_0\sigma_0^2 + n_1\sigma_1^2}{n_0+n_1} + \left(\frac{\mu_1-\mu_0}{n_0+n_1}\right)^2 \cdot n_0 \cdot n_1$

Varc of combn $= \frac{1}{n_0+n_1}\left(n_0\sigma_0^2 + n_1\sigma_1^2 + \frac{(\mu_1-\mu_0)^2}{\frac{1}{n_0}+\frac{1}{n_1}} \right)$: say $n_0 = n_1 = n$: results $\frac{1}{2n}\left(n\sigma_0^2 + n\sigma_1^2 + \frac{(\mu_1-\mu_0)^2}{2} n\right)$

$= \frac{1}{2}\left(\sigma_0^2 + \sigma_1^2 + \frac{(\mu_1-\mu_0)^2}{2}\right)$

This same effect: T. extra contribution due to choice of mean, should be $\left(\frac{(\mu_1-\mu_0)}{2}\right)^2$; since this amount is added to every σ^2.

So it seems to check.

$\frac{n_0\sigma_0^2}{n_0+n_1}$

.08 $= \frac{n_0 n_1}{n_0+n_1}\left(\frac{\sigma_0^2}{n_1} + \frac{\sigma_1^2}{n_0} + \left(\frac{\mu_1-\mu_0}{n_0+n_1}\right)^2 \right) = \frac{1}{\frac{1}{n_1}+\frac{1}{n_0}}\left(\frac{\sigma_0^2}{n_1} + \frac{\sigma_1^2}{n_0} + \frac{(\mu_1-\mu_0)^2}{n_0+n_1} \right)$

205.18 : T. measurement of a quantity n_0+n_1 times: This gives t. product of 2 Gsn dfs. — or t. addition of exponents of e.

In ALP, when we combine various models, we use additive (linear) wtng of their d.f.'s.

[65 sequencal v.s. parrallel measurements.

If pirallal combn: t. linearly wtd. sum of 2 Gaussians w. diffrnt μ's σ's is not a Gaussian.

.33 To answer 205.01! We want to extend Mexm to more general situations; not just considering another linear coif. This is more like t. present F.T. problem; I want to do predn. of a gn. driven fund. I have several possl. drivers. Say I decide on linear predn. How many & which drivers should I use for min expected future error?

.37 → A possl. soln. to .33: Say I have 20 possl. drivers & 1 driven. Consider t. coding of t. entire corpus of 21 time series. This is, in general, t. correct way to do probs of this sort. (is this t. same as 202.23?) → 208.01

A somewhat different problem from 206.33: More like t. AAE$_{valn.}$ problem:

We have several "Actions" that we've tried in t. past, & we have their outcomes: From this we have a problem in ~~extrapolation~~ ~~understanding~~ understanding (or ordering) sets extrapolation (like Q-A$_i$ extrapolation. — ~~we get a pd. for~~ given an arby new Q', to get a p.d. for all poss. A's. "Operator induction" is one way to look at it. Or, construction of a "stochastic (say, in which ass'ts are (Q,A) pairs. Or t. general sequential predictor, in which t. corpus has form is $(Q_1 \diamond A_1, \Delta, Q_2 \diamond A_2, \Delta \cdots Q_n \diamond A_n)$: the Δ's are "stop symbols", & t. form of t. generator makes each Q_i, A_i have a probty that's indep of t. previous Q,A's in t. corpus (i.e., it's essentially a stoch. lang.).

.05

T. lang. (.05 ff in particular, enables us to make it time dependent) would seem to give a joint d.f. ~~of~~ on all possl. AA's. From this, we ask t. SOY question. —

What is t. best choice of AA, & how Good can we expect it to be — & what is t. distribn. function of ~~the~~ of that "Goodness"? AA's??

.07

A possl. Q w.r.t. Jürgen's LHL: We have th. various AA's of t. past, & their empirical "yields" — we have a few "ΔR" values, but we are interested in t. long-term future mean $\frac{\Delta R}{\Delta T}$. We don't have past examples of that! — Well, also, J's goal is not so clear: The idea of maximizing max ~~mean future~~ ΔR for (unknown) life of T.M. seems ill-defined. — Also, there is some possy that J. really doesn't want that Goal! → See 188.27-.40

189.13 data after problem for TM.

Re: Optzn: Say we have a set of [AA$_i$]: We have their past yield histories. From this (~~in~~ theory) we obtain a d.f. of yield for t. "next try" (whatever that means!). We can then apply t. SOY sdn. method to t. resultant joint Prob'y Distribn to obtain SOY — but even w.o. that, (soysoln.) we can get Expected "next try" yield for each AA$_i$, & chose t. max.

How to express this as a "well defined OZ problem"? First, we don't have $\Sigma C_i = \infty$ for t. various AA$_i$ P.D.'s. So how best to spend our "time"? — how much time to spend ~~on each~~ computing t. p.d. of out put of each AA$_i$? Say the yields Y_i are linearized Utilitys. Say Y_i^j $(j=1/m)$ are m possl. values for Y_i, t. yield of AA$_i$. At t. end of time t, we will have $p_i^j(t)$ — t. total no. of codes from AA$_i$ yielding Y_i^j — for Expected yields $\sum_j p_i^j(t) \cdot Y_i^j$.

Can we ask (as an OZ problem) to use our time T so that one of the i has as large $\sum_j p_i^j \cdot Y_i^j$ as possl? This could bias our results toward those Y_i^j for which one could get large $Y_i^j p_i^j$ quickly! — Not nearly t. best choice!

.01 Well: O.K. for a certain short amt of time, one gets spuriously by $Y_i^j P_i^j$ — but after that time one has to ↑ t. P's at other i values.

A possy: That one's prime goal should be Max $\sum_i \sum_j P_i^j$ Then one picks i ∋ $\sum_j P_i^j Y_i^j$ is Max

With finite CB, its easy to bias t. results toward certain P_i^j's.

This is true even in computing simple probability ratios — (i.e. t. proby that t. next bit in a seq. will be 0 rather than 1. — If one's goal is to max $P(1)+P(0)$, then this would seem "to be unbiasd"

Still: I'm not sure that t. OZ goal of obtaining one i ∋ $\sum_j P_i^j Y_i^j$ is max, is wrong.

[More exactly: find i ∋ t. proven lower bnd on ⟶ is Max.

The Pr.z ↑ Q is still in limbo:

Sept 10:) credit.
Sept 25
1.800 8917770 ← info on Pr.z.
RCH

.12 : 206.40: A modifn of Mexm: Say we want to predict a particular T.S.

We have an ∞ of T.S's to use for regressn for predn. We order these Time series in

.14 by their correln, say w. t. predicted T.S. So each T.S. has an integer no. betw. 1 & ∞.

We then want to code specifications of k-tuples of these TS's. Say we had t.p.c. of specification of each k tuple. Then each k tup has a predn w. σ^2, obtaind via Mexm linearly (i.e. $\sigma^2 \frac{N+k}{N-k}$) — & we "combine them" using t. pc's of the specifications of each k tup.

8:47 clck (osso) 8:50 10:00

So t. problem seems to be getting a reasonable p.c. for k tup. specifn.

.22 There may be another problem with the possl. ranges of each coef being ±∞.

This (& other of t. forgo problems) also occurs in ordinary linear regressn.

I think it may be solvd .22: Make range ±R. Get pd. for tomorrow's price as a functn of R. Then let $R \to \infty$. This may work if we only consider 1 value of k — but I think t. wt. of k+1 tups v.s. k tups is like $\frac{1}{R}$. (!). — T. idea is that a particular k tup: (Say k = 2) will have a wt ∝ the hy pc. region of a 2R × 2R square. The wt. will be t. ratio of t. area of t. hy p.c. region to t. area of t. square. — which is ∝ $\frac{1}{R^2}$ as $R \to \infty$. If we have a 3 tup, its wt will go as $\frac{1}{R^3}$ so we see t. ratio of wts of 3 tup to 2 tup will ∝ $\frac{1}{R}$.

One could deal w. this by having the R for t. k tup ↑ slower as k ↑; so that R^k was t. same for all k. Say $R_k = f(R, k)$:

& $\lim_{k\to\infty} R = \infty$: R_k^k is indep of k. $= R_1$

∴ $R_k = (R_1)^{1/k}$

It mite be possl. to build that R^k effect into t. apriori p.c. assignment for t. k-tups (←??)

Th. forgo problem occurs in ordinary linear regressn. I think in my original formulation of Mexm, I did not get a good way to ↓ pc as I ↑ t. no. of coefs. I later devised that method that assumed t. seq. was not exponentially increasing... (This is a bit too extreme, but apriori should ↓ very rapidly as exponential rate ↑.) — T. idea was

Treat t. roots of t. char. eq. should be not outside t. unit ⊙ ... but actually I probly want to allow them outside, but ↓ their pc rapidly as they move away from th unit ⊙. I had lots of trouble interpreting t. original eqns — It may conceivably be easier if I just allow all values of ↓ pc rapidly w. "radius" of a root. (radius = "amplitude"). — Maybe Gaussian d.f.? If Gaussian, then most sets of coeffs would be in or near t. unit [complex] h. sphere — So they'd have about t. same amplitude.

.06 → Another way of thinking about linear regression (& also t. ktup prodn. method): In linear regression, once we tell how many coeffs are to be used, [illegible] and how far back t. cov. matrix window goes (or whether it is for x window), then we start to code consists only of the [illegible] Normally distributed errors. These are to be used to code these errors can also be obtained from t. same R window or x window [illegible] averages of past errs. This is for something like a "steady state" stationary sequence.

The .06 needs more explanation: Continuing: It means that once we specify t. no. of coeffs., all we have to do is give the errors — coded wrt a Normal distribution of known σ. So, essentially, all we have to worry about is t. p.c. of "t. no. of coeffs": we can use $\frac{1}{n}$ with some upper bnd on n or Rissi's "universal" d.f. or whatever: — it may make little difference.

Looks v. G. — I think G. Wallace used this.

— That way, work for linear regression — but assigning pc's to ntups may be much different.

A possl. way: Assign an integer to each t.s. eg 208.12 — .14 To decb. a k tuple: Give in comma-free notation, t sequence of k integers that define t. k tup. Do this in order (lowest no. first, highest last). If P(n) is t. p.c. assoc. w. coding integers ($n = 1|00$), then if N(i) is t. integer assoc. w. t. i^{th} t.s. in this k tup ($i = 1|k$) then $\left[\prod_{i=1}^{k}\right]$ [redacted] $P(N(i)+1-i)$] mult by p.c. of an end symbol, ist. pc of [redacted] for k tup.

or use "P(n)" not specific about what approx. to use for P(n).

P(n) is, of course, a non-recursive function, we will use various approxns

We use $N(i)+1-i$ because t. first [illegible] t.s. has pc = P(N(1)) The second has pc = P(N(2)−1) because the integer N(1) (which is < N(2)) is no longer a possl., so the codes for all integers > N(1) become more likely — Or, — well it better than that! →

.31 Instead: After we code N(1), we just need to code N(2)−N(1), & this will be 1 or more. Next we have to code N(3)−N(2), etc.

Considering .31 ff, we may want to give very hy prob to small integers. (Not so obvious — maybe not so!)

---

p.c. of "end" symbol: well, we can code the k tup by first giving k, then t. following k integers: so k+1, comma-free integer codes

Result is that changing fr tup from k to k+1 has pc of $< \frac{k}{k+1}$; If the k+1 tuple will cost at least P(1) for an additional t.s. specifier, and it could be as long as P(r), where r is t. hy order of one of t. T.S. of 208.12 — .14

This assumes $P(n) \simeq \frac{1}{n} \cdots$ which, of course, diverges

→ See 210.20 for more intelligible discussn.