FROM GAME THEORY TO DISTRIBUTED OPTIMIZATION AND BACK AGAIN

David H. Wolpert

NASA Ames Research Center

http://ic.arc.nasa.gov/~dhw/

November, 2003

NASA-ARC-TR-03-117

PRODUCT DISTRIBUTION (PD) THEORY

An extremely simple formalism, with applications throughout the physical and social sciences:

- **i) Bounded rational game theory**
- **ii) Optimization, constrained or not, over *any* measurable space**
- **iii) Closed loop control**
- **iv) Reinforcement learning**
- **v) Sampling of distributions**
- **vi) High-dimensional numeric integration**

Especially suited to distributed applications

TALK ROADMAP - JUST A TASTE ...

1) **Review noncooperative game theory**

2) **Discuss its three major shortcomings**

3) **Present the information-theoretic extension of game theory to allow bounded rationality - PD theory**

4) **Discuss PD theory's advantages (cf. (2)), and its relation to statistical physics**

5) **Relate PD theory - and therefore game theory - to distributed optimization**

NONCOOPERATIVE GAME THEORY

- **A set of N *players*, each choosing a *pure strategy*, $z_i \; \square \; \square_i$**
- **A set of N *cost functions* $h_i(z)$; player i wants minimal h_i**
- **However i's cost depends on the joint-strategy - and i can only choose its own strategy**

Example: Prisoner's dilemma cost table ($h_1(z)$, $h_2(z)$)

	Player 2's strategy	
Player 1's strategy:	**(2, 2)**	**(10, 0)**
	(0, 10)	**(7, 7)**

Other examples: rock-paper-scissors, chess, war, diplomacy

- **Given a game, what joint strategy do we expect to get?**
- **Hypothesis: Player i chooses whatever strategy is best for it, given the strategies of the other players, $z_{(i)}$, :**

$$\text{argmin}_{z_i}\ h_i(z_i, z_{(i)})$$

z is a *Nash equilibrium* iff for all players i, for all z'_i,

$$h_i(z'_i, z_{(i)}) \leq h_i(z_i, z_{(i)})$$

Example: Prisoner's dilemma, Nash equilibrium indicated by {}

	Player 2's strategy	
Player 1's strategy:	(2, 2)	(10, 0)
	(0, 10)	{7, 7}

- ***Problem***: **Some games have no Nash equilbrium**

Example: Modified prisoner's dillema $(h_1(z), h_2(z))$

		Player 2's strategy	
Player 1's strategy:		(2, 2)	(3, 7)
		(0, 10)	(7, 7)

- ***Solution***:

i) **Players take *mixed strategies* $P_i(z_i)$**

ii) **Each player i picks P_i to minimize expected cost given mixed strategies of others, $P_{(i)}$:**

$$E(h_i) = \int dz_i\, P_i(z_i)\, E(h_i \mid z_i)$$

ii) **$\{P_i(z_i)\}$ is a Nash equilibrium iff for all players i, no change to $P_i(z_i)$ decreases $E_{P_i|P_{(i)}}(h_i)$**

- ***Unresolved problems*:**

  1) Finding Nash equilibria is a (hard) multi-criteria optimization problem

  2) In real world, players are never fully rational, due to limited computational power, if nothing else.

Bounded rationality

  3) More than one Nash equilibrium, in general.

- **Attempts to date to solve (2) are just more elaborate models of (human) players**

  - Underlying problem is arbitrariness of the models.

Combining Information theory and game theory/

- **Say we know the strategies of the players other than i, and we know i's expected cost given those $P_{(i)}$:**

  Then Maxent says it is "most conservative" to estimate that P_i is the minimizer of

  $$L_i(P) = \square E(h_i) - S(P)$$

  with *S* the Shannon entropy.

- **Given the entropy over i's possible strategies, predict that P_i minimizes i's expected cost:**

  Then again, P_i minimizes $L_i(P)$

BOUNDED RATIONALITY

- **At Nash equilibrium, each P_i separately minimizes**

$$E(h_i) \;=\; \square dz\; h_i(z)\; \square_j\; P_j(z_j)$$

- **Allow broader class of goals (*Lagrangians*) for the players**

Example

i) Each P_i separately minimizes the Lagrangian

$$L_i(P) \;=\; \square E(h_i) \;-\; S(P)$$

for some appropriate function *S* (e.g., entropy)

ii) $\square <$ is bounded rationality

Some of bounded rationality's advantages and implications/

1) To have player i optimize several cost functions $\{h_{ij}\}$ at once, replace

$$E(h_i) = \int dz\, h_i(z)\, P(z)$$

$$\Downarrow$$

$$\sum_j a_j \left[\int dz\, h_{ij}(z)\, P(z)\right]^2$$

in all Lagrangians, where a_j is a probability distribution

- This is *not* the same as replacing the cost function with an average cost function
- Biases toward P that are good for many $\{h_{ij}\}$, rather than P that are great for one h_{ij} and bad for the others

2) Choose $S(q) = \square_i \square dz_i S_i(P_i(z_i))$.

Then bounded rationality is identical to conventional, full rationality — every player wants to minimize expected cost. Only now there is a new cost function:

$$f_i(z, P_i) = \square h_i(z) - S_i(P_i(z_i)) / P_i(z_i)$$

$-S_i(P_i(z_i)) / P_i(z_i)$ **measures the cost to player i for calculating** $P_i(z_i)$

3) In a *team game*, all h_i equal the *world utility*, G

E.g., "social welfare function", $G(z) = \sum_i g_i(z)$

(So a single objective function is being minimized, in a distributed fashion...)

- Under general conditions on *S*, *L*(P) is a convex surface with a single global minimum. In particular, this is true for *S* being the Shannon entropy:

• One and only one solution
• The solution is easy to find

Combining statistical physics and game theory/

- **Jaynes showed that all statistical physics ensembles arise from minimizing**

  $$L_i(P) = \square E(h_i) - S(P),$$

  with S the Shannon entropy

- **Mean field theory arises if P is a product distribution; bounded rational game theory = mean field theory (!)**

Much of the mathematics of statistical physics can be applied to bounded rational game theory

The grand canonical ensemble can model bounded rational games in which the number of actors varies.

***Intuition*: Players with "types" = particles with properties**

Example 1 (microeconomics):

i) **A set of bounded rational companies,**
ii) **with cost functions given by market valuations,**
iii) **each of which must decide how many employees of various types to have.**

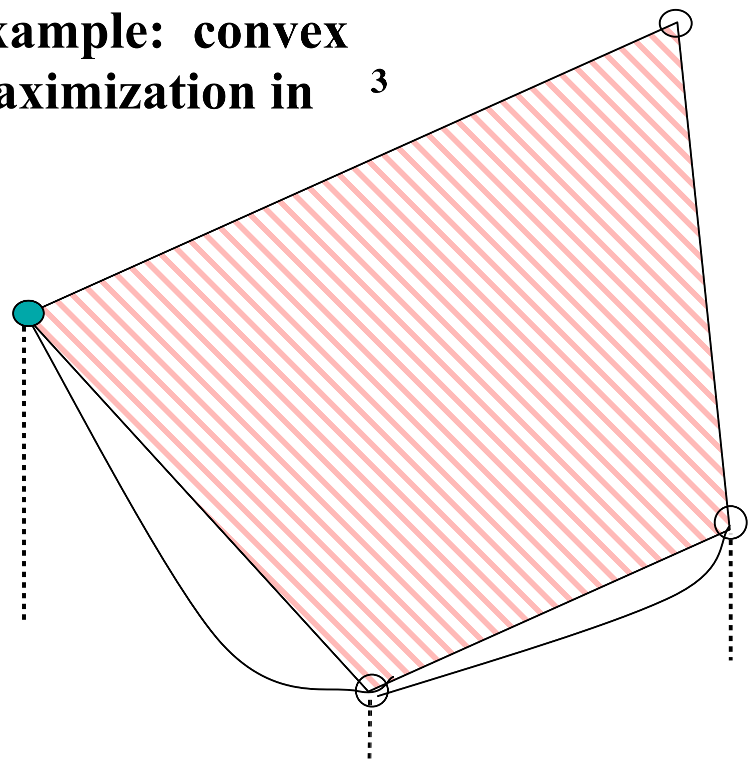
Example 2 (evolutionary game theory):

i) **A set of species,**
ii) **with cost functions given by fractions of total resources they consume,**
iii) **each of which must "decide" how many phenotypes of various types to express.**

OPTIMIZATION

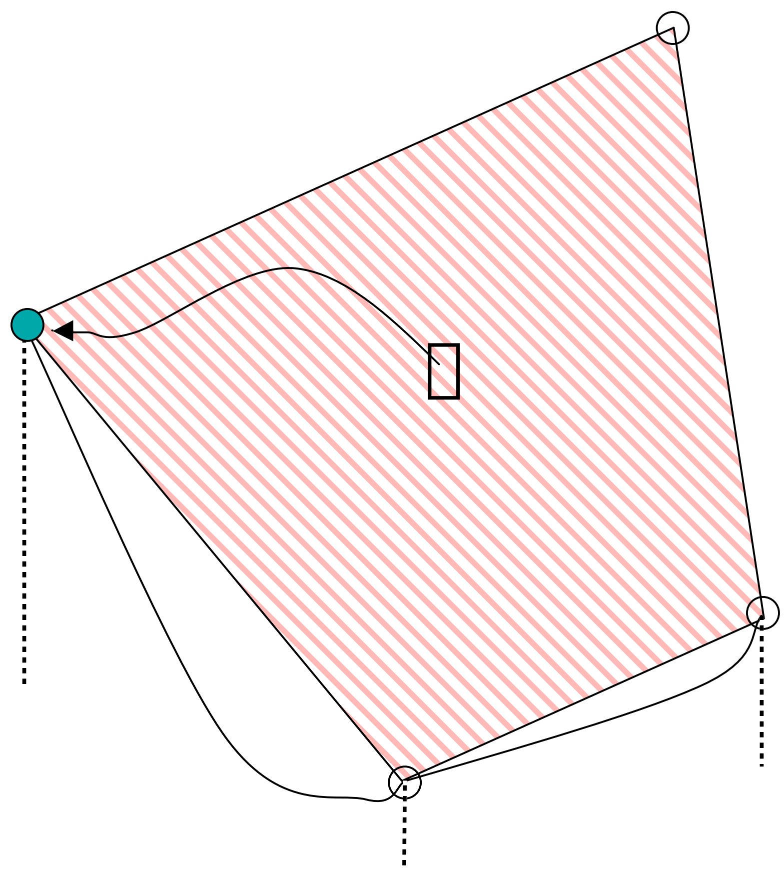
- ***Core issue:*** **how to use information at one point to choose a next sample point.**
- **NP hard is when such information is useless.**

Example: convex maximization in ^{3}



- **Why optimization (and therefore control, high-dimension integration, etc.) can be hard**

- **Best case is continuous domains, where smoothness can be exploited — if you aren't trapped in a vertex**
- **So: Distort problem so solution is off the border, and then weaken the distortion.**

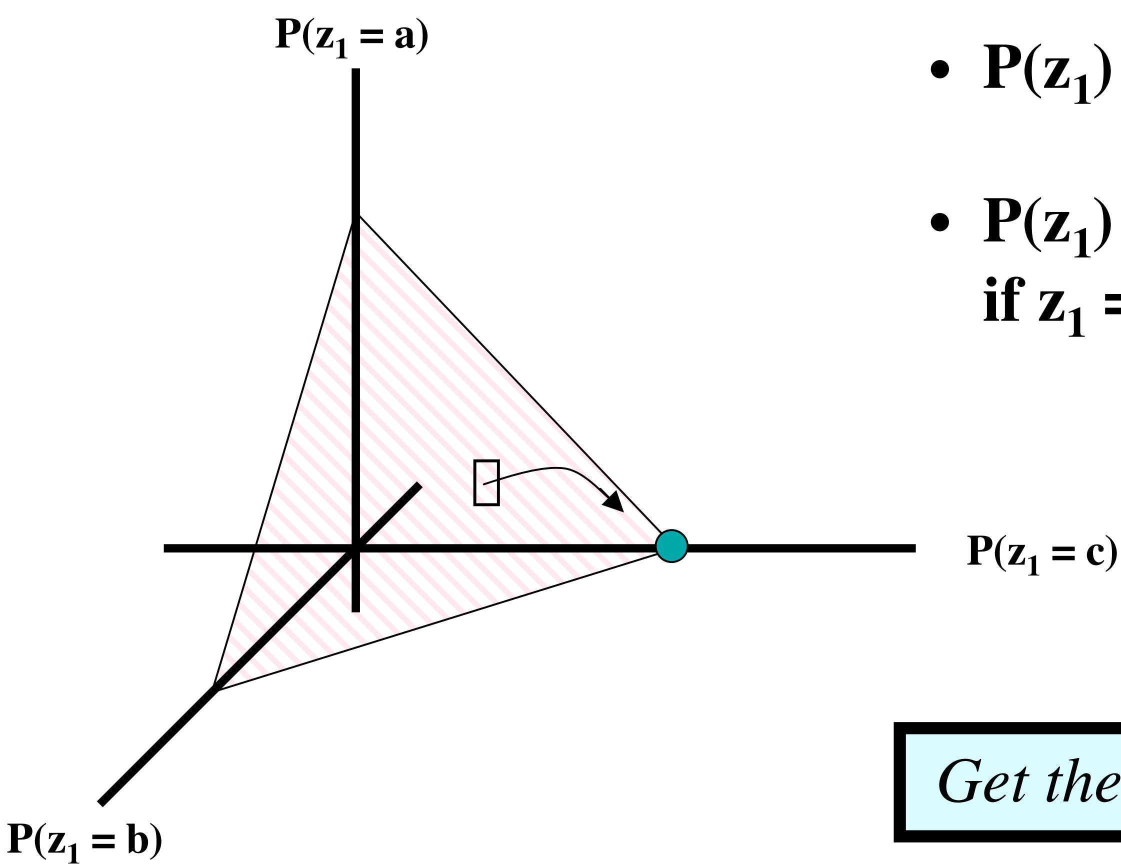


▯ — distorted problem solution

● — original problem solution

- **Example: Interior point methods**

- **Can do this for discrete domains by using a probability distribution as the continuous variable**



- **$P(z_1) \in \Re^3$**

- **$P(z_1) = \delta(z_1 - (1, 0, 0))$ if $z_1 = c$ exactly**

Get the solution off the border

1) **For each successive distorted problem, exploit smoothness to search over P(z)'s**

- **Gradient descent, Newton's method . . . even simulated annealing.**

Gradient descent to optimize <u>categorical</u> variables subject to <u>categorical</u> constraints

2) **Example: To minimize G(z), find the P(z) minimizing**

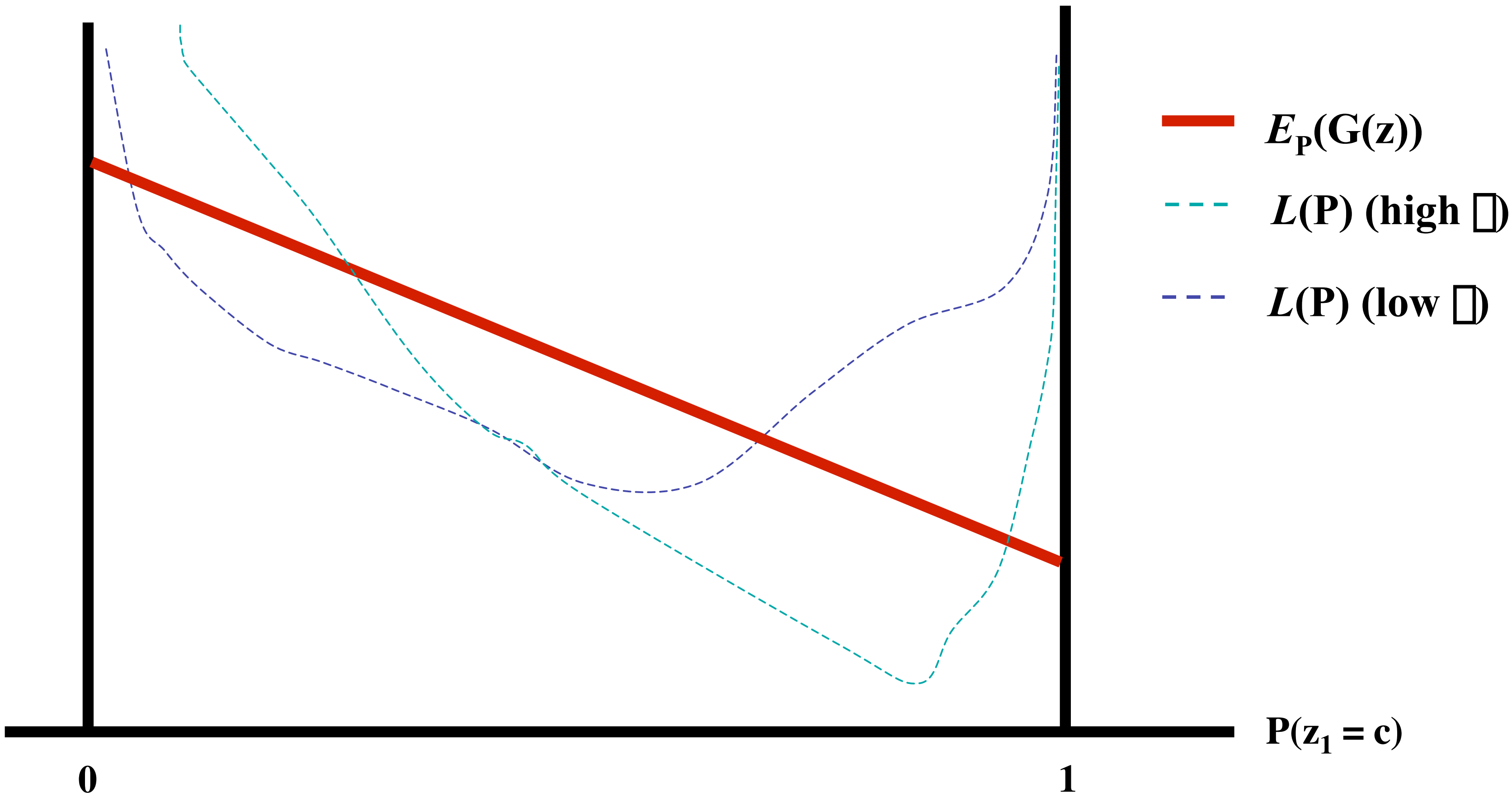
$$L(\mathrm{P}) = \square E(\mathrm{G}) - S(\mathrm{P})$$

- **Larger □ = less distortion — *anneal***

$E_P(G) = \int dz\, G(z)P(z)$ is linear in P(z). Therefore,

If -*S*(P) is convex, so is *L*(P)

So if the derivative of S(P) is infinite at the simplex border, *L*(P) has a unique minimum, off that border



Example: Take S(P) to be the Shannon entropy,

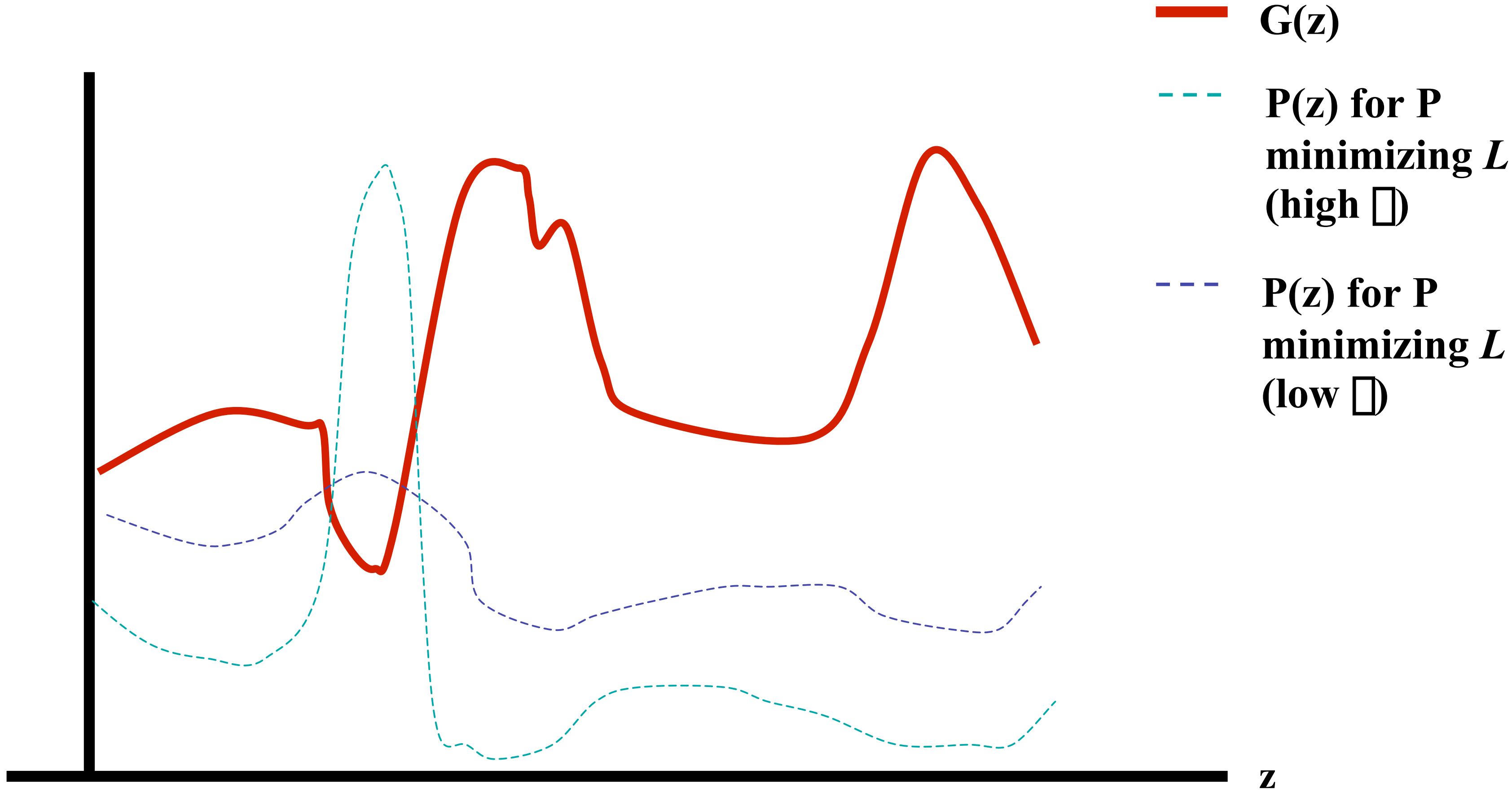
$$S(P) = -\int dz\, P(z) \ln[P(z)]$$

- **As required, -S(P) is convex, with infinite derivative at the simplex border**

- **L(P) is minimized by the *Boltzmann distribution*,**

$$P(z) \propto exp(-\beta\, G(z))$$

As □□ , P(z) becomes a delta function about the z minimizing G(z)



Simulated annealing:

1) At each β, perform an associated Metropolis-Hastings random walk

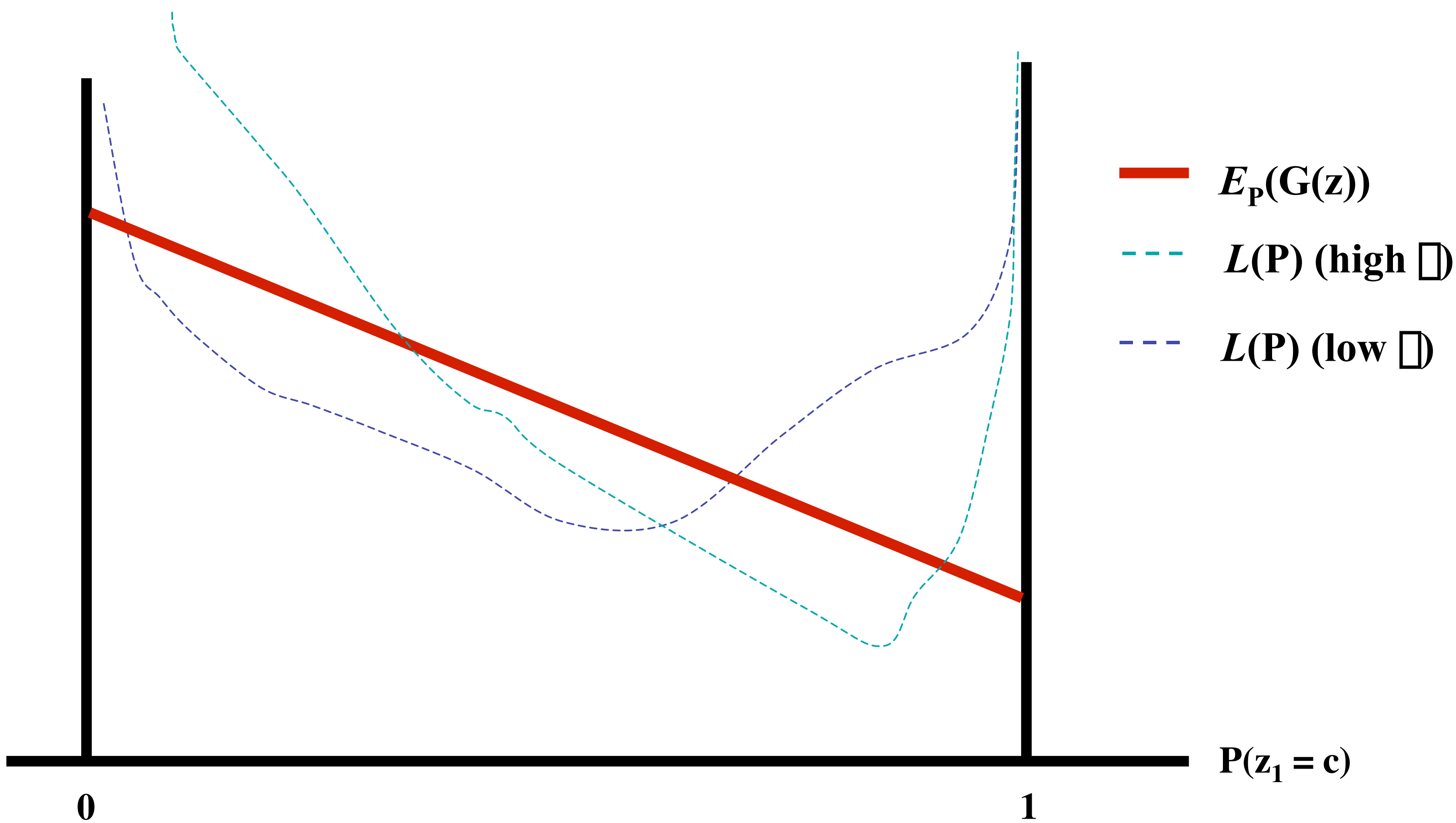
2) That walk eventually gives a random sample of $P_{\beta}(z)$

3) When you think it has, increase β, and repeat

So when you get to high β, your sample is likely to be close to argmin G(z)

. . . **inefficient**

Alternative: Use gradient descent (for example) to find P(z) at each :



P(z) lives in a ***huge*** space. How parameterize it?

With a distributed parameterization, parameters can be estimated separately from each other. So optimization

i) can be *parallelized*,
ii) can be used for *distributed control*,

So . . .

Use a product distribution: $$P(z) = \prod_i P_i(z_i)$$

Life isn't *precisely* perfect:

- $$L(q) = \square E_P(G(z)) - S(P)$$
  $$= \square dz\, G(z)\ \square_i P_i(z) - S(P)$$

- L is linear in P — but multilinear in the P_i

- So even for convex S(P), L(P) need not be convex:

  At any □, L(P) can have multiple minima

- Even for entropic S,

  At any □, P(z) can have multiple peaks

(just like multiple Nash equilibria . . .)

TAKE-HOME MESSAGE:

Whenever you encounter a distribution $P(z)$ that is difficult to deal with, try expanding it as a product distribution

$$\prod_i P_i(z_i)$$

with associated Lagrangians.