۱)

آ) طبق تعریف عامل عقلانی درک محیط را با سنسورهایش بدست می آورد

و با عوامل محرک بر روی محیط اثر می گذارد (با عوامل محرک درک انجام نمی کنند) (نادرست)

ب) طبق تعریف دوباره یکی از عوامل عقلانی بودن دانش پیشین است

چون هرچه اطلاعات بیشتر باشد عملکرد درست تر و بهتری انجام خواهد شد (درست)

ج) اولین علت اینکه با توجه به مثال نقض کلاس ممکن است در sideways ها

به ملاقی برویم که جواب ندارد و در لوپ قرار بگیریم.

علت دوم بررسی احتمال:

Random:

$$P = 0{,}14 \Rightarrow E(\text{برد}) \overset{\text{توزیع هندسی}}{=} \frac{1}{P} \approx 7 \begin{cases} \xrightarrow{\text{۶ شکست}} 6 \times 3 = 18 \quad (\text{تعداد مراحل برای باخت}) \\ \xrightarrow{\text{ابرد}} \underline{4} \times 1 = 4 \quad (\text{تعداد مراحل برای برد}) \end{cases} \Rightarrow 22$$

sideways:

$$P(\text{باخت})\, F(\text{باخت}) + P(\text{برد})\, F(\text{برد}) = 0{,}06 \times 64 + 0{,}94 \times 21 \simeq 23{,}6$$

(درست)

د) زمانی که هیچگاه کاهش نیابد یعنی می تواند ثابت هم باشد و اگر پیال صفر داشته باشیم

می تواند در حلقه قرار گیرد و تعداد گام مهم نیست و در نتیجه به جواب نمی رسد (نادرست)

هـ) با توجه به اینکه در simulated annealing فقط حالت فعلی را داریم و به گذشته ربطی ندارد مقدار

حافظه همواره ثابت است

در این الگوریتم در در local قرار بگیریم خارج می شویم چون بصورت رندوم خارج می شود فقط آ نباید سریعتر از

$C/\log n$ افت کند در الگوریتم ژنتیک هم اگر در لوکال باشیم احتمال صفر نیست و باعث خروج می شود

(درست)

۲)

الف)

حالت‌ها: موقعیت قرارگیری هر کاوشگر بصورت مختصات آن

کنش‌ها: حرکت در چهار جهت یا ثابت ماندن برای هر کاوشگر به شکل مستقل

هزینه هر کنش: اگر حداقل یک کاوشگر حرکت کند ۱ در غیر اینصورت ۰

هدف: رسیدن همه به سطر آخر به طوری که کاوشگری که ابتدا در $(1, i)$ بوده به $(n-i+1, n)$ برسد.

ب)

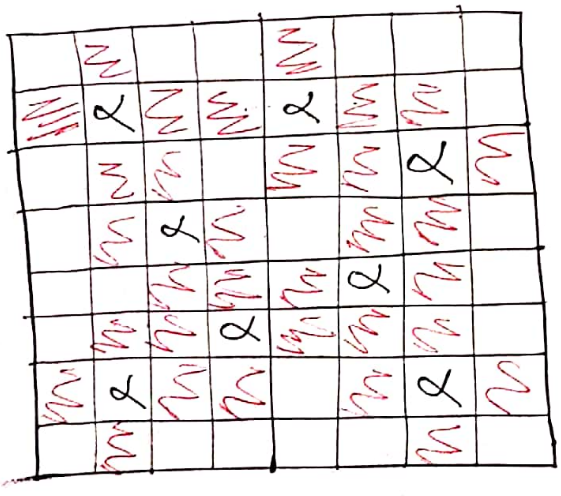
n^2 خانه داریم که هر کاوشگر در یکی می‌تواند باشد و دو کاوشگر هم نمی‌توانند در یک خانه باشند

پس برای اولی n^2 انتخاب بعدی n^2-1 و در نهایت برای n امی، n^2-n انتخاب داریم

$$\Rightarrow O = n^2 \times (n^2-1) \times (n^2-2) \cdots \times (n^2-n) = \frac{n^2!}{(n^2-n)!}$$

$$\Rightarrow O\left((n^2)^n\right) = O\left(n^{2n}\right)$$

ج) بدترین حالت این است که هر کاوشگر بتواند هر ۵ حرکت خود را بکند در این حالت تعداد حالات 5^n می‌شود که خیلی بزرگ است



مثال برای $n = 8$

$\boxed{\text{branching factor} < 5^n}$

د) برای این تابع هیوریستیک فاصله اقلیدسی دو نقطه را در نظر می گیریم که بصورت زیر تعریف می شود.

$$h = |n - i - x_i| + |n - y_i|$$

این تابع قابل قبول است چون مینیمم فاصله دو نقطه است و از هر مسیری که برویم کمتر از مینیمم نمی شود
اگر کاوشگرهای دیگر هم حضور داشته باشند باز هم همچنان مینیمم فاصله همین است و حداقل این مقدار باید طی شود البته می تواند هر بیشتر شود که شرط قابل قبول بودن تابع ما را نقض نمی کند چون minimum است.

ه) $\checkmark \min(h_1, \dots, h_n) \longrightarrow$ چون برای اینکه حداقل یکی به مقصد برسد مینیمم فاصله h_i است و بین همه ی توابع مینیمم را انتخاب کنیم پس در واقع داریم نزدیک ترین فاصله نزدیک ترین نقطه به مقصد را در نظر می گیریم که همواره درست است

$\times \sum_i h_i \longrightarrow$ نادرست است چون در هر مرحله می توانند کاوشگرها همزمان حرکت کنند و در زمان h_i به جواب برسند مثلا هر کدام در سطر درست و ستون $n-1$ باشند با 1 حرکت به مقصد می رویم ولی تابع می گوید n.

$\times n \min(h_1, \dots, h_n) \rightarrow$ این حالت دقیقا مثل قبل و مثال آن برقرار است

$\checkmark \max(h_1, \dots, h_n) \rightarrow$ باتوجه به اینکه می خواهیم همگی به مقصد برسند پس انتخاب ماکسیمم درست است چون ماکس که انتخاب می شود حداقل در آن تعداد حرکت به مقصد می رسد که اینجا نا کی ما است پس درست است

$\times n \max(h_1, \dots, h_n) \longrightarrow$ دقیقا مثل ۳ مثال پیش مثال نقض دارد

$\checkmark \frac{\sum_i h_i}{n} \longrightarrow \sum_i h_i < n \max(h_i) \Rightarrow \frac{\sum_i h_i}{n} < \max(h_i)$ چون کمتر از ۲ حالت قبلی است پس این هم درست است.

و)

حالت‌ها: موقعیت قرارگیری هر کاوشگر

کنش‌ها: حرکت در ۴ جهت فقط برای یک کاوشگر یا ثابت ماندن (حالت عوض نمی‌شود)

هزینه هر کنش: یک واحد برای تغییر هر حالت یا صفر برای ثابت ماندن

ب) مانند قبل است و همان $O(n^{2n})$

ج) در بدترین حالت هرکدام از n کاوشگر ۵ حالت دارند پس

$$\text{branching factor} < 5 \times \binom{n}{1} = \boxed{5n}$$

۳)

الف) برای قابل قبول بودن در هر خانه مقدار مینیمم تا مقصد را حساب می‌کنیم و باید h_1 و h_2 از این کمتر مساوی باشد

min A : ۱۳ min B : ۱۲ min C : ۱۱ min D : ۸

min E : ۲ min F : ۵ min G : ۰

h_1 و h_2 کمتر مساوی است پس قابل قبول است

برای یکنوا بودن از $h(a)-h(b) \leqslant c(a,b)$ استفاده می‌کنیم

A و B : h_2 → ۱۲-۱۰ ≤ ۱ ✗ ، h_1 → ۹٫۵-۹ ≤ ۱ ✓

h_2 حذف می‌شود و یکنوا نیست پس فقط با h_1 ادامه می‌دهیم

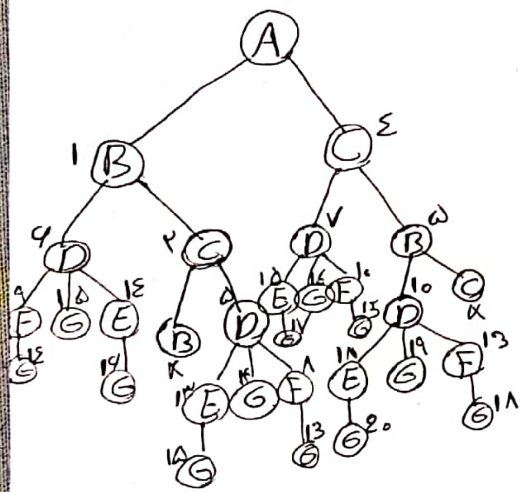
A و C : ۱٫۵ ≤ ۴ و B و C : ۱ ≤ ۱ ، B و D : ۲ ≤ ۵ ، C و D : ۱ ≤ ۳

D و E : ۵٫۵ ≤ ۸ ، D و F : ۳ ≤ ۳ ، D و G : ۷ ≤ ۹ ، F و G : ۴ ≤ ۵ و E و G : ۱٫۵ ≤ ۴

پس h_1 یکنوا است

ب)

گراف در ادامه رسم شده است



: A-B-D-G

DFS : می‌تواند ، BFS : می‌تواند چون نزدیک است

هزینه یکنواخت : نمی‌تواند چون ABDG زودتر است

جستجو با h_1 : ابتدا B انتخاب می‌شود پس C سپس D سپس F و سپس G و به جواب می‌رسیم

جستجو با h_2 : ابتدا B سپس C سپس D سپس F و G و به جواب می‌رسیم

A-C-D-G

DFS : می‌تواند BFS : می‌تواند برای ۳ حالت بعدی مثل حالت قبل نمی‌شود

A-B-C-D-F-G

DFS : می‌تواند BFS : نمی‌تواند چون عمق زیاد است برای h_1 و h_2 می‌شود

	A B D G	A C D G	A B C D F G
DFS	✓	✓	✓
BFS	✓	✓	✗
uniform	✗	✗	✓
h_1	✗	✗	✓
h_2	✗	✗	✓

ج)

با توجه به قسمت الف که مینیمم برای B 12 است مقدار h_3 کمتر مساوی است

پس $\boxed{0 \leq h_3(B) \leq 12}$

د) برای یکنوا بودن با همسایه ها بررسی می کنیم مقدار آن را x فرض می کنیم

$$\left.\begin{array}{ll} A \text{ و } B: & |10-x| \leq 1 \Rightarrow 9 \leq x \leq 11 \\ B \text{ و } C: & |9-x| \leq 1 \Rightarrow 8 \leq x \leq 10 \end{array}\right\} \Rightarrow \boxed{9 \leq x \leq 10}$$

ه)

برای اینکه ابتدا C انتخاب شود یک شرط داریم

$$4 + h_3(C) \leq 1 + h_3(B) \Rightarrow 13 \leq 1+x \Rightarrow \boxed{12 \leq x}$$

برای اینکه بعد C ، B انتخاب شود یک شرط زیر داریم

$$5 + h_3(B) \leq 7 + h_3(D) \Rightarrow 5+x \leq 7+7 \Rightarrow \boxed{x \leq 9}$$

پس این اتفاق نمی افتد و تهی است.

۴)

الف) n جای خالی و n کلمه داریم برای کلمه اول n جای خالی برای بعدی n-1 و ...

فضای حالت $= n \times (n-1) \times \dots \times 1 = \boxed{n!}$

ب) برای تعریف همسایه می توان به صورت اینکه هر دو جمله‌ای که دو کلمه آن ها جابجا شده است همسایه هستند

برای مثال برای "هرچه زودتر خواهش می کنم آزاد کنید من را" می توان "هرچه زودتر آزاد می کنم خواهش کنید من را"

ج) بله چون در هر حالت یک استیت را داریم و مقدار معنا داری آن را می دانیم و می توانیم معنا داری همسایه‌های هر استیت را بدون نیاز به استیت های قبلی محاسبه کنیم و در نهایت به جواب برسیم

(ممکن است در مینیمم محلی گیر کنیم و قطعا به نتیجه نمی رسیم)

با اضافه کردن randomwalk و randomrestart می توانیم از مینیمم محلی خارج شویم و به احتمال بیشتری به جواب نهایی می رسیم.

د) برای استفاده از الگوریتم ژنتیک ابتدا ۴ حالت random ایجاد کنیم برای حالت ابتدایی حال از این ۴ حالت ۴ حالت را بصورت رندوم انتخاب می کنیم که می تواند تکراری هم باشد جمله ما ۸ کلمه دارد پس می توانیم به عنوان مثال دسته‌ها را به دو قسمت ۳ و ۵ تایی تبدیل کنیم حال برای crossover هم هر دو زوج را بخش هایشان را جابجا می کنیم در نهایت هم می توانیم اگر عضو تکراری داریم با mutation آن را تغییر دهیم.

۵)

آ)

$$f(x) = \|x\|_۲^۲ = x_1^2 + x_2^2 + x_3^2 + \dots + x_n^2 \Rightarrow \nabla f = \frac{\partial f}{\partial x_1}\hat{x}_1 + \frac{\partial f}{\partial x_2}\hat{x}_2 + \dots + \frac{\partial f}{\partial x_n}\hat{x}_n$$

$$= (۲x_1, ۲x_2, ۲x_3, \dots, ۲x_n) = \boxed{۲x}$$ جواب

$$f(x) = \|Ax\|_۲^۲ = \sum_{i=1}^{n} ((Ax)_i)^2 = \sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij}x_j\right)^2$$

$$\Rightarrow \nabla f = \frac{\partial f}{\partial x_k} = \frac{\partial}{\partial x_k}\sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij}x_j\right)^2 = \sum_{i=1}^{m}\frac{\partial}{\partial x_k}\left(\sum_{j=1}^{n} a_{ij}x_j\right)^2$$

$$= \sum_{i=1}^{m} ۲a_{ik}\sum_{j=1}^{n} a_{ij}x_j = \sum_{i=1}^{m} ۲a_{ik}(Ax) = ۲\sum_{i=1}^{m} a_{ik}(Ax) = \boxed{۲A^TAx}$$ جواب

$$f(x) = \|Ax-b\|_2^2 + \gamma\|x\|_2^2 \Rightarrow \nabla f = \nabla\|Ax-b\|_2^2 + \nabla\gamma\|x\|_2^2$$

$$\Rightarrow \nabla\|Ax-b\|_2^2 = \frac{\partial}{\partial x_k}\|Ax-b\|_2^2 = \frac{\partial}{\partial x_k}\sum_{i=1}^{m}((Ax)_i - b_i)^2 = \frac{\partial}{\partial x_k}\sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij}x_j - b_i\right)^2$$

$$= \frac{\partial}{\partial x_k}\sum_{i=1}^{m}\left(\left(\sum_{j=1}^{n} a_{ij}x_j\right)^2 - ۲b_i\sum_{j=1}^{n} a_{ij}x_j + b_i^2\right) = \sum_{i=1}^{m}\frac{\partial}{\partial x_k}\left(\left(\sum_{j=1}^{n} a_{ij}x_j\right)^2 - ۲b_i\sum_{j=1}^{n} a_{ij}x_j + b_i^2\right)$$

$$= \underbrace{\sum_{i=1}^{m}\left(\frac{\partial}{\partial x_k}\left(\sum_{j=1}^{n} a_{ij}x_j\right)^2\right)}_{\text{قسمت قبل}} + \underbrace{\sum_{i=1}^{m}\frac{\partial}{\partial x_k} \times -۲b_i\sum_{j=1}^{n} a_{ij}x_j}_{=۰ \text{ if } k\neq j} + \underbrace{\sum_{i=1}^{m}\frac{\partial}{\partial x_k} b_i^2}_{۰}$$

$$= ۲A^TAx - \sum_{i=1}^{m} -۲b_i a_{ik} + ۰ = ۲A^TAx - ۲A^Tb = \boxed{۲A^T(Ax-b)}$$

$$\Rightarrow \nabla f = \boxed{۲A^T(Ax-b) + \gamma ۲A^TAx}$$

جواب

بـ ۱)

$f(x) = \|x\|_2^2$

$$x_0 \longrightarrow \nabla = 2x_0 \longrightarrow x = x_0 - \alpha \nabla f = x_0 - 2\alpha x_0 = (1-2\alpha)x_0$$

این کار را چند بار انجام می دهیم تا همگرا شود یعنی از x_0 فرضی شروع می کنیم و مدام x را آپدیت می کنیم تا به مینیمم همگرا شود

$f(x) = \|Ax\|_2^2$

$$x_0 \longrightarrow \nabla = 2A^TAx_0 \longrightarrow x = x_0 - \alpha \nabla f = x_0 - 2\alpha A^TAx_0 = (1-2\alpha A^TA)x_0$$

مانند قبلی چند بار انجام می دهیم تا همگرا شود

$f = \|Ax-b\|_2^2 + \gamma\|x\|_2^2$

$$x_0 \longrightarrow \nabla = 2A^T(Ax_0-b) + 2\gamma A^TAx_0 \longrightarrow x = x_0 - \alpha \nabla f = x_0 - \alpha 2A^T(Ax_0-b) - 2\alpha\gamma A^TAx_0$$

مانند قبل تکرار می شود تا همگرا شود .

۲)

$g(x) = \|x\|_2^2 = x^Tx$

شرط محدب بودن : $g(\alpha x_1 + (1-\alpha)x_2) \leqslant \alpha g(x_1) + (1-\alpha)g(x_2)$

$$\Rightarrow g(\alpha x_1 + (1-\alpha)x_2) \leqslant \alpha x_1^Tx_1 + (1-\alpha)x_2^Tx_2$$

$$\Rightarrow (\alpha x_1 + (1-\alpha)x_2)^T(\alpha x_1 + (1-\alpha)x_2) \leqslant \alpha x_1^Tx_1 + (1-\alpha)x_2^Tx_2$$

$$\Rightarrow (\alpha x_1^T + (1-\alpha)x_2^T)(\alpha x_1 + (1-\alpha)x_2) = \alpha^2 x_1^Tx_1 + (1-\alpha)^2 x_2^Tx_2 + \alpha(1-\alpha)(x_1^Tx_2 + x_2^Tx_1)$$

$$\Rightarrow 0 \leqslant (\alpha-\alpha^2)x_1^Tx_1 + (\alpha-\alpha^2)x_2^Tx_2 - (\alpha-\alpha^2)(x_1^Tx_2 + x_2^Tx_1)$$

$$\Rightarrow 0 \leqslant (x_1^T + x_2^T)(x_1 + x_2) \Rightarrow 0 \leqslant (x_1+x_2)^T(x_1+x_2)$$ عبارت همواره درست

پس فرض ما بصورت بازگشتی درست است .

د)

$$g(x) = f(Ax - b)$$

شرط محدب بودن: $g(\alpha x_1 + (1-\alpha)x_2) \leqslant \alpha g(x_1) + (1-\alpha) g(x_2)$

$$\Rightarrow f\big(A(\alpha x_1 + (1-\alpha)x_2) - b\big) \leqslant \alpha f(Ax_1 - b) + (1-\alpha) f(Ax_2 - b)$$

$$\Rightarrow f\big(\alpha(Ax_1 - b) + (1-\alpha)(Ax_2 - b)\big) \leqslant \alpha f(Ax_1 - b) + (1-\alpha) f(Ax_2 - b)$$

$$\left.\begin{array}{l} Ax_2 - b = y_2 \\ Ax_1 - b = y_1 \end{array}\right\} \Rightarrow f(\alpha y_1 + (1-\alpha) y_2) \leqslant \alpha f(y_1) + (1-\alpha) f(y_2)$$

شرط محدب بودن f که همواره درست است پس فرض درست است

وبازگشتی ثابت می شود

ه)

$$\left.\begin{array}{l} g(x) = \|x\|_2^2 \xrightarrow{\text{طبق قسمت ج محدب}} \\ f(x) = \|Ax - b\|_2^2 = g(Ax - b) \xrightarrow{\text{طبق قسمت د محدب}} \end{array}\right\} \Rightarrow f(x) \text{ محدب است}$$