دانشکده‌ی مهندسی برق

بهینه‌سازی محدب ۱ — مدرس: دکتر یاسایی

تمرین کامپیوتری سری دوم

نام و نام‌خانوادگی : خشایار غفاری — شماره دانشجویی: ۹۸۱۰۰۲۱۵

پرسش ۱   برآورد درست‌نمایی بیشینه

آ)

$$l(\lambda_t) = e^{-\lambda_t}\frac{\lambda_t^{N_t}}{N_t!} \Rightarrow log(l(\lambda_t)) = -\lambda_t + N_t log(\lambda_t) - log(N_t!)$$

به دنبال λ_t می گردیم که تابع $log(l)$ را بیشینه کند.
تابع $log(l)$ به وضوح مشتق‌پذیر و مقعر است زیرا تابع لگاریتم، مقعر می باشد و $-\lambda_t$ نیز مقعر است و عبارت دیگر نیز ثابت است. پس طبق این توضیحات نقطه‌ی بیشینه‌ی این تابع در $\lambda_{t_{(MLE)}}$ رخ می‌دهد اگر و تنها اگر داشته باشیم: $log(l)'(\lambda_{t_{(MLE)}}) = 0$

$$\Rightarrow -1 + \frac{N_t}{\lambda_{t_{(MLE)}}} = 0 \implies \lambda_{t_{(MLE)}} = N_t$$

در حالت $N_t = 0$ نیز مشتق همین عبارت بالا می‌شود و جواب فرقی ندارد.
ب) تعداد دفعاتی که نوعی رویداد در بازه‌ی یک ساعته رخ می‌دهد در ساعت‌های مختلف مستقل از یکدیگر هستند. پس داریم:

$$\begin{aligned} l(\lambda_1, \lambda_2, \ldots, \lambda_{24}) &= \mathbb{P}((N_1 = N_1) \cap (N_2 = N_2) \cap \cdots \cap (N_{24} = N_{24})) \\ &= \mathbb{P}(N_1 = N_1)\mathbb{P}(N_2 = N_2)\ldots\mathbb{P}(N_{24} = N_{24}) = \prod_{t=1}^{24} e^{-\lambda_t}\frac{\lambda_t^{N_t}}{N_t!} \\ \implies log(l(\lambda_1, \lambda_2, \ldots, \lambda_{24})) &= -\sum_{t=1}^{24}\lambda_t + \sum_{t=1}^{24} N_t log(\lambda_t) - \underbrace{\sum_{t=1}^{24} log(N_t)}_{\text{ثابت}} \end{aligned}$$

پس هدف ما بیشینه کردن تابع زیر می‌باشد.

$$-\sum_{t=1}^{24}\lambda_t + \sum_{t=1}^{24} N_t log(\lambda_t) - \rho(\sum_{t=1}^{23}(\lambda_{t+1} - \lambda_t)^2 + (\lambda_1 - \lambda_{24})^2)$$

و بنابراین باید تابع زیر را کمینه کنیم:

$$\sum_{t=1}^{24}\lambda_t + (-\sum_{t=1}^{24}N_t log(\lambda_t)) + \rho(\sum_{t=1}^{23}(\lambda_{t+1}-\lambda_t)^2 + (\lambda_1-\lambda_{24})^2)$$

که به راحتی دیده می‌شود هر جمله‌ی آن محدب است.
پ) وقتی $\rho \to \infty$ هزینه‌ای که هر کدام از جملات $(\lambda_{t+1}-\lambda_t)^2$ به تابع هزینه اضافه می‌کنند به ∞ میل می‌کند. (چون همگی نامنفی هستند.) پس تابع هزینه وقتی کمینه می‌شود که تمام این جملات صفر باشند. پس خواهیم داشت:

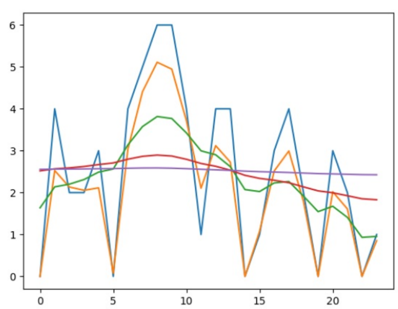
$$\lambda_1 = \lambda_2 = \cdots = \lambda_{24}$$

پس تابع هزینه‌ی ما به این شکل خواهد شد:

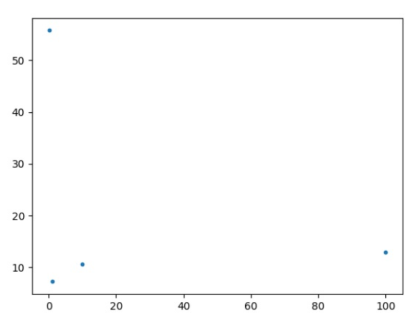
$$\sum_{t=1}^{24}\lambda_t + (-\sum_{t=1}^{24}N_t log(\lambda_t)) = 24\lambda_1 - log(\lambda_1)\sum_{t=1}^{24}N_t \xRightarrow{\text{مشتق برابر صفر}} \lambda_1 = \frac{\sum_{t=1}^{24}N_t}{24}$$

ت)

شکل ۱: نمودار λها بر حسب ρ



همانطور که انتظار داشتیم با زیاد شدن ρ مقادیر ما از N فاصله می‌گیرند و به طرز مشهودی هموارتر می‌شوند و در $\rho = 100$ تقریبا یک خط صاف داریم یعنی مقادیر λ_iها خیلی نزدیک به هم شدند.
ث)



در این نمودار قرینه‌ی مقدار لگاریتم درست‌نمایی را بر حسب ρهای مختلف می‌بینیم که برای $\rho = 1$ از بقیه کمتر است پس انتخاب این مقدار بهترین است.

پرسش ۲ سطوح فعالیت بهینه

آ) مساله‌ی ما به شکل زیر می‌باشد.

$$\begin{aligned} \text{maximize} \quad & \sum_{j=1}^{n} r_j(x_j) \\ \text{s.t} \quad & x \succeq 0 \\ & Ax \preceq c^{max} \end{aligned} \tag{۱}$$

توجه کنید که $r_j(x_j) = min\{p_j x_j,\ p_j q_j + p_j^{disc}(x_j - q_j)\}$
و با ثابت نگه داشتن x جواب مساله‌ی LP

$$\begin{aligned} \text{maximize} \quad & \mathbf{1^T y} \\ \text{s.t} \quad & \text{diag}(p)x \succeq y \\ & \text{diag}(p)q + \text{diag}(p^{disc})(x - q) \succeq y \end{aligned} \tag{۲}$$

جایی است که $\text{diag}(p)x = y$ یا $\text{diag}(p)q + \text{diag}(p^{disc})(x - q) = y$ و نمی‌تواند از هیچکدام بزرگ‌تر باشد پس همان $\sum_{j=1}^{n} r_j(x_j)$ است. پس در نهایت مساله‌ی LP معادل با مساله‌ی ما مساله‌ی زیر است:

$$\begin{aligned} \text{maximize (over } x,\ y) \quad & \mathbf{1^T y} \\ \text{s.t} \quad & \text{diag}(p)x \succeq y \\ & \text{diag}(p)q + \text{diag}(p^{disc})(x - q) \succeq y \\ & x \succeq 0 \\ & Ax \preceq c^{max} \end{aligned} \tag{۳}$$

ب)

$$x_{opt} = [3.99999996 \ \ 22.49999989 \ \ 30.99999995 \ \ 1.50000005]$$
$$average = [-3. \quad -1.44444445 \quad -4.48387097 \quad -6.00000001]$$

پرسش ۳ برنامه ریزی بهینه سرعت وسیله نقلیه

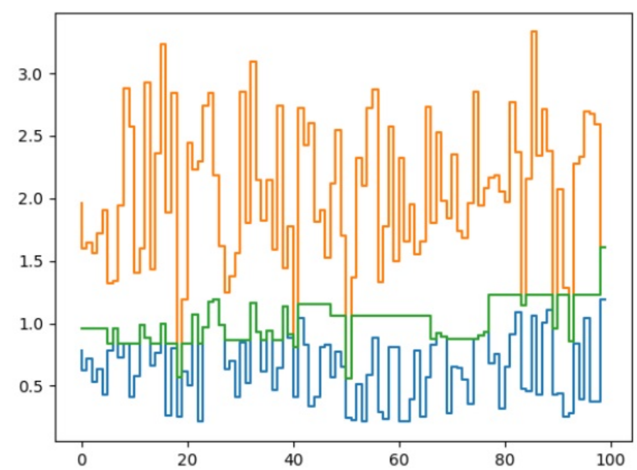
آ) همانگونه که در سوال مطرح شد، پارامتر $\frac{d_i}{s_i}$ زمان طی‌شده در بخش iـ‌ام را نشان می‌دهد پس تعریف می‌کنیم:

$$t_i := \frac{d_i}{s_i}$$
$$\Longrightarrow \text{مجموع سوخت مصرفی} = \sum_{i=1}^{n} \Phi(s_i)\frac{d_i}{s_i} = \sum_{i=1}^{n} \Phi(\frac{d_i}{t_i})t_i$$

پس باید تابع $\sum_{i=1}^{n} \Phi(\frac{d_i}{t_i})t_i$ را کمینه کنیم که چون هر جمله‌ی آن تابع پرسپکتیو بر حسب (d_i, t_i) است پس تابعی محدب است. قیدهای مساله به شکل زیر هستند.

$$i \text{ سرعت در بخش} = s_i = \frac{d_i}{t_i} \Longrightarrow s_i^{min} \leq \frac{d_i}{t_i} \leq s_i^{max} \xRightarrow{\text{ثابت } d_i} \frac{d_i}{s_i^{max}} \leq t_i \leq \frac{d_i}{s_i^{min}}$$
$$i \text{ زمان رسیدن به بخش} = \sum_{j=1}^{i} t_j \Longrightarrow \tau_i^{min} \leq \sum_{j=1}^{i} t_j \leq \tau_i^{max}$$

که هر دو قید بیان‌شده بر حسب t خطی هستند پس قیدها محدب هستند.
پس با حل مساله‌ی معرفی‌شده بر حسب t می‌توان s بهینه را پیدا کرد.
ب) در حالت بهینه مصرف سوخت ۲۶۱۷ کیلوگرم می‌باشد.



پرسش ۴ بازنویسی قیود

آ)

```
    x = cp.Variable(1)
    y = cp.Variable(1)
    objective = cp.Maximize(0)
    constraints = [cp.inv_pos(x) + cp.inv_pos(y) <= 1]
    prob = cp.Problem(objective, constraints)
    print(prob.is_dcp())

```

زیرا توابع $\frac{1}{x}$ و $\frac{1}{y}$ در حالت کلی محدب نیستند و حتما باید دامنه تابع مثبت باشد باید از دستور cp.inv-pos استفاده بکنیم.

ب)

```
    x = cp.Variable(1)
        y = cp.Variable(1)
        objective = cp.Maximize(0)
        constraints = [x >= cp.inv_pos(y)]
        prob = cp.Problem(objective, constraints)
        print(prob.is_dcp())

```

زیرا تابع xy در حالت کلی محدب نیست و چون در اینجا دامنه‌ی ما مثبت است می‌توانیم یکی را به سمت دیگر نامساوی ببریم تا تابع محدب شود.

پ)

```
    x = cp.Variable(1)
        y = cp.Variable(1)
        objective = cp.Maximize(0)
        constraints = [cp.quad_over_lin(x + y, cp.sqrt(y)) <= x - y + 5]
        prob = cp.Problem(objective, constraints)
        print(prob.is_dcp())

```

در تمرین‌های تئوری دیده بودیم که تابع $\frac{f^2}{g}$ وقتی که تابع f نامنفی و محدب باشد و تابع g مثبت و مقعر باشد، محدب است. در این سوال نیز این شرایط برای تابع $\frac{(x+y)^2}{\sqrt{y}}$ برقرار است لذا می‌توانیم از دستور استفاده کنیم.

ت)

```
    x = cp.Variable(1)
        y = cp.Variable(1)
        z = cp.Variable(1)
        objective = cp.Maximize(0)
        constraints = [x + z <= 1 + cp.geo_mean(cp.vstack([x - cp.
   quad_over_lin(z,y), y]))]
        prob = cp.Problem(objective, constraints)
        print(prob.is_dcp())

```

داریم: $\sqrt{y(x-\frac{z^2}{y})} = \sqrt{xy-z^2}$. که کسر $\frac{z^2}{y}$ مشابه قسمت قبل است.