مجله ریاضیات کاربردی واحد لاهیجان، سال ششم، شماره ۲۳، زمستان ۸۸، ص ص ۷۲-۶۹

بررسی روش زوتندیک در مسایل برنامه ریزی خطی

حسین نورمحمدی*

گروه ریاضی، دانشگاه آزاد اسلامی واحد انار

چکیده

یکی از مهمترین مسایل علم ریاضی برنامه ریزی خطی و کاربردهای آن می باشد. برای حل این گونه مسایل الگوریتم های نقطه درونی از سال ۱۹۸۴ مورد استفاده قرار گرفته اند. در این مقاله سعی می شود ایدهٔ زوتندیک در حالت خطی مورد تجزیه و تحلیل قرار گیرد. اساس کار این الگوریتم شروع از یک نقطه اکیداً درونی در ناحیه شدنی و حرکت در جهت گرادیان تابع هدف است. حسن این روش و به طور کلی روش های نقطه درونی عدم نیاز به جواب های شدنی پایه ای برای شروع می باشد.

کلمات کلیدی: الگوریتم های نقطه درونی، برنامه ریزی خطی، ایدهٔ زوتندیک

۱ مقدمه

اگر چه هنوز هم روش سیمپلکس یکی از کاراترین روش های حل مسایل برنامه ریزی خطی محسوب می شود، اما این روش نمی تواند مرتبه پیچیدگی چند جمله ای را برای یافتن جواب بهینه تضمین کند. به طور متوسط در یک مساله با m قید، تعداد تکرارهای لازم جهت رسیدن به جواب بهینه $1/5\,m$ است[۳].

اما کلی و مینتی (۱۹۷۲) دسته مسایلی را ارایه دادند که تعداد تکرارهای لازم برای رسیدن به جواب بهینه در مورد آن ها بسیار بیشتر از عدد فوق بود[۶]. در (۱۹۸۴) الگوریتم کارمارکار که دارای مرتبه پیچیدگی چند جمله ای بود، برای حل این گونه مسایل ارایه شد[۴]. این الگوریتم سرآغاز دستهٔ بزرگی از الگوریتم ها گردید که معمولاً به نام الگوریتم های نقطه درونی شناخته می شوند. شروع کار این الگوریتم ها عموماً از یک نقطه شدنی است که این نقطه در فاز اول این روش ها به دست می آید. حرکت در جهت بردار گرادیان تابع هدف یک ایدهٔ کلی برای حل مسایل برنامه ریزی خطی و غیر خطی است که توسط زوتندیک مورد بررسی قرار گرفت. این ایده در مسایل برنامه ریزی خطی به حرکت در جهت بردار c منتهی می شود. بنابراین بردار c در پیدا کردن جهت حرکت نقش اساسی را خواهد داشت.

*عهده دار مکاتبات

آدرس الکترونیکی: *h_noormohamad@yahoo.com*

در اینجا گذری به مفاهیم پایه ای که در ادامه مورد استفاده قرار می گیرند، خواهیم داشت.

در این مقاله یک مساله برنامه ریزی خطی به صورت زیر در نظر گرفته می شود.

$$Max\ \ Z = c^t x$$
$$s.t : Ax \le b$$

که در آن c برداری n بعدی و b برداری m بعدی و ماتریس A ، $(m \times n)$ بعدی خواهد بود. در نظر داشته باشید که قیود نامنفی را می توان به فرم بالا در مجموعه محدودیت ها گنجاند.

قضیه ۱ جهت گرادیان محدودیت $a^i x \le b_i$ به طرف خارج ناحیه شدنی است و گرادیان تابع هدف بهترین جهت بهبود دهنده برای یک مساله LP به فرم بالاست[۲].

تبصره اگر مساله دارای جواب بهینه متناهی باشد حرکت در جهت c تا رسیدن به نقطه بهینه ادامه می یابد و اگر ناحیه شدنی مساله بیکران باشد حرکت در این جهت می تواند به طور نامحدود ادامه یابد که در این حالت مقدار تابع هدف به صورت نامحدود افزایش می یابد.

قضیه ۲ برادر e_m همواره یک نقطه شدنی برای مساله برنامه ریزی خطی زیر خواهد بود:

$$Max\ \ \ Z = c_1x_1 + c_2x_2 + \ldots + c_nx_n - Mx_{n+1}$$
$$s.t : a_1x_1 + a_2x_2 + \ldots . + a_nx_n + (b - Ae_n - e_m)x_{n+1} \le b$$

که در آن منظور از بردار e_m برداری m بعدی است که همه مولفه های آن یک می باشند[۷].

۲ تشریح روند حرکت

در ابتدا با شروع از نقطه شدنی x° جهت c را برای حرکت در نظر می گیریم. تا به نقطه ای مانند x^1 روی مرز ناحیه شدنی برسیم. جهت جدید حرکت را به گونه ای می یابیم که بر بردار گرادیان ابر صفحه های فعال در نقطه جدید عمود باشد و با بردار c زاویه کمتر از $\frac{\pi}{2}$ بسازد. در صورتی که تعداد این ابر صفحه ها بیش از n-1 باشد، تعداد n-1 تای آن ها را بوسیله محک ارایه شده در گام دوازدهم الگوریتم انتخاب می کنیم. با ادامه این روند به یک نقطه راسی می رسیم که معمولاً همان جواب بهینه است و یا به آن نزدیک خواهد بود. از این گام به بعد حرکت الگوریتم از یک نقطه راسی به یک نقطه راسی جدید خواهد بود که مقدار تابع هدف در آن نقطه بهتر است. نقطه ترک فضای شدنی نقطه بهینه می باشد. این الگوریتم بعد از در نظر گرفتن حالات خاص به صورت زیر خواهد بود. برنامه الگوریتم تحت نرم افزار maple اجرا شده است.

۳ مراحل الگوریتم

۱) شروع

۲) d:=c

۳) s={}

۴) $i := 0$

۵) N:=Ad

۶) M:=b – A x^i

۷) اگر $M \geq 0$ نبود آن گاه $x^{(i-1)}$ جواب مساله است و برو به ۱۷.

۸) $\lambda := \min\left\{\frac{M_j}{N_j} : N_j > 0, j \notin S\right\} = \frac{M_{k_i}}{N_{k_i}}$

۹) $S := SUk_i$ (اگر λ یکتا نبود آن گاه همهٔ k_i ها را به s اضافه می کنیم).

۱۰) اگر λ موجود نبود آن گاه مساله دارای جواب نامتناهی است، برو به ۱۷.

۱۱) $x^{i+1} := x^i + \lambda d$

۱۲) {حداکثر n-1 اندیسی مانند j که $j \in s$ و مقدار $\frac{a^j d}{\|a^j\|\|d\|}$ مثبت ترین باشد } $S :=$

۱۳) $d := solve(cd = 1, a^j d = 0 : j \in S)$

۱۴) اگر d موجود نبود مساله دارای بینهایت جواب است، برو به ۱۷.

۱۵) $i := i + 1$

۱۶) برو به مرحله (۵)

۱۷) پایان

مثال

$$\begin{aligned}
&\max \quad Z = -x_1 + x_2 + 2x_3 \\
&s.t : -3x_1 - 4x_2 - 3x_3 \geq 1 \\
&-5x_1 - 4x_2 - 3x_3 \leq 10 \\
&7x_1 + 4x_2 + 9x_3 \leq 18 \\
&x_1 \leq 0, x_2 \geq 0
\end{aligned}$$

حل: با شروع از نقطه $x^0 = (-1,1,-1)$ در سه تکرار اول به نقاط $x^1 = (-\frac{8}{7}, \frac{8}{7}, -\frac{5}{7})$ ، $x^2 = (-\frac{9}{7}, \frac{205}{56}, -\frac{5}{7})$, $x^3 = (-\frac{9}{2}, 0, \frac{25}{6})$ می رسیم که x^3 نقطه بهینه با تابع هدف $\frac{77}{6}$ می باشد. برای حل مساله از روش سیمپلکس احتیاج به ۵ متغیر اضافی و ۵ تکرار داریم.

٤ نتیجه گیری و پیشنهاد

در این مقاله سعی شده است که بر خلاف روش های نقطه درونی که از رسیدن به مرز ناحیهٔ شدنی پرهیز می شود، بعد از تکرار اول، کاملاً روی مرز حرکت کرد. چون بعضی از مولفه های N در محاسبات دخیل نیستند، شاید بتوان آن ها را از ابتدا از مساله حذف کرد و بدین ترتیب کارایی الگوریتم را بالا برد. ممکن است روند این الگوریتم را بتوان همانند روش سیمپلکس به صورت جدولی در آورد.

منابع

[1] I. ADLER, N. KARMARKAR, M.G.C. RESENDE, AND G. VEIGA, An Implementation of Karmarkar's Algorithm for Linear programming' , mathematical programming, 44:297-335,1989.

[2] M.S.BAZARAA, J.J. Jarvis and H. Sherali, 'Linear Programing and Network Flows, 2nd ed. 'John willey and sons, New York, 1990.9.

[3] H. A. Eislet, C.-L. Sandblom, 'Linear programming and its Applications' 'springer-Verlag Berlin Heidelberg, 2007.

[4] N.K.Karmarkar , 'A New polynomial-time Algorithm for Linear Programming' ,combinatorica, 4:373-395, 1984.

[5] L.G. Khachiyan, 'A polynomial Algorithm for Linear programming' ,soviet Math. Dokl., 20:191-194,1979.

[6] Robert J. Vanderbei, 'Linear Programing : Foundations and Extentions' 2nd edn. Kluwer, Boston,MA , 2001.

[7] C.Roos, T. Terlaky, J.-PH. Vial, 'Interor point Methods for Linear Optimization, '2nd edn. Springer , New York, NY, 2006.