حل مساله تخصیص تعمیم یافته به روش شبکه عصبی

مریم اعتمادی*
عضو هیئت علمی دانشگاه آزاد اسلامی مرکز فومن
etemadi.maryam@gmail.com

چکیده

در این مقاله روش شبکه عصبی را برای حل مساله تخصیص تعمیم یافته به کار می بریم. این مساله تعمیم مساله معروف تخصیص است که به صورت یک مساله برنامه ریزی صحیح صفر و یک فرموله می شود. برای حل این مساله با استفاده از روش شبکه عصبی ابتدا آن را به یک مساله برنامه ریزی غیر خطی تبدیل می کنیم سپس دو نوع ساختار شبکه عصبی یکی بر اساس روش تابع جریمه و دیگری روش ضرایب لاگرانژین افزوده را برای آن به کار برده و با هم مقایسه می کنیم. ملاحظه می شود که شبکه عصبی بر مبنای روش تابع جریمه برای مسایل بهینه سازی ترکیبی مناسب نیست زیرا یا به جواب نشدنی می رسد و یا در یک جواب شدنی غیر بهین متوقف می گردد در حالی که شبکه عصبی بر مبنای روش ضرایب لاگرانژین افزوده تا حدی این مشکل را بر طرف می کند.

کلمات کلیدی: شبکه عصبی، مساله تخصیص، برنامه ریزی صحیح، روش تابع جریمه، روش ضرایب لاگرانژین افزوده.

۱ مقدمه

مساله تخصیص تعمیم یافته (GAP)[1] یک مساله بهینه سازی ترکیبی Np-hard است که در آن هدف مینیمم کردن هزینه تخصیص مجموعه ای از کارها به افراد می باشد به طوری که هر کار دقیقا به یک فرد داده شود اما هر فرد می تواند چند کار انجام دهد. این مساله دارای کاربردهایی مانند تخصیص کار به کامپیوترهای موجود در یک شبکه کامپیوتری، طراحی شبکه های ارتباطات، تعیین مسیر وسایل نقلیه و ... دارد. مساله GAP تعمیم مساله تخصیص(AP)[2] می باشد که مساله ای معروف در تحقیق درعملیات بوده و برای آن الگوریتم با زمان چند جمله ای وجود دارد. در دهه های گذشته روش شبکه عصبی برای حل بلادرنگ[3] مسایل بهینه سازی ترکیبی بزرگ مقیاس مورد توجه قرار گرفته است. در حال حاضر بیشتر ساختارهای شبکه عصبی بر مبنای روش

[1] Generalized Assignment Problem
[2] Assignment Problem
[3] Real time

* عهده دار مکاتبات

تابع جریمه هستند برای مثال شبکه عصبی هاپفیلد این گونه است. این نوع شبکه عصبی دارای خواص خوبی در همگرایی و پایداری است اما دارای مشکل انتخاب مناسب پارامتر جریمه است. اگر پارامتر جریمه کوچک باشد آنگاه ممکن است جواب نهایی نشدنی شود. از سوی دیگر اگر پارامتر جریمه بزرگ باشد آنگاه هر جواب شدنی می تواند به یک نقطه جذب تبدیل شود. دراین مقاله دو نوع ساختار شبکه عصبی یکی بر اساس روش تابع جریمه و دیگری بر اساس روش ضرایب لاگرانژین افزوده بیان می کنیم. ابتدا مساله برنامه ریزی صحیح صفر و یک را به مساله برنامه ریزی غیر خطی معادل آن تبدیل نموده. سپس روش تابع جریمه و روش ضرایب لاگرانژین افزوده را برای تبدیل مساله به یک مساله بدون محدودیت به کار می بریم و سیستم های دینامیک معادلات دیفرانسیل آن ها را تشکیل داده و جواب های حاصل از این دو روش را مقایسه می نماییم.

۲ مساله تخصیص تعمیم یافته

مدل ریاضی مساله تخصیص تعمیم یافته برای n مصرف کننده و m مرکز سرویس به صورت زیر است:

$$\textbf{(GAP)}\quad Min \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

s.t.

$$\sum_{j=1}^{n} a_{ij}x_{ij} \le b_i, \qquad i \in I = \{1,2,\cdots,m\}, \qquad (۱)$$

$$\sum_{i=1}^{m} x_{ij} = 1, \qquad j \in J = \{1,2,\cdots,n\}, \qquad (۲)$$

$$x_{ij} \in \{0,1\}, \qquad \forall i \in I, j \in J.$$

که در آن c_{ij} هزینه تخصیص مصرف کننده j به مرکز سرویس i، a_{ij} مقداری از ظرفیت کاری مرکز سرویس i که برای مصرف کننده j مصرف می شود و b_i ظرفیت کاری مرکز سرویس i است. متغیر x_{ij} برابر یک است اگر مصرف کننده j به مرکز سرویس i تخصیص داده شود و در غیر این صورت صفر است. مجموعه I مجموعه اندیس های مربوط به مراکز سرویس و J مجموعه اندیس های مربوط به مصرف کننده ها است. به محدودیت های (۱) محدودیت های کوله پشتی و به محدودیت های (۲) محدودیت های تخصیص گفته می شود. در واقع وجود محدودیت های کوله پشتی باعث تغییر مساله تخصیص به تخصیص تعمیم یافته می گردد. چون مدار VLSI شبکه عصبی می تواند یک مساله بزرگ مقیاس را در زمان بسیار کوتاهی حل کند لذا روش شبکه عصبی برای حل مسایل بزرگ مقیاس مورد توجه قرار گرفته است.

۳ روش شبکه عصبی

محققین زیادی روش شبکه عصبی را برای انواع متفاوتی از مسایل بهینه سازی مورد مطالعه قرار داده اند. به طور کلی این روش شامل سه مرحله اصلی است. در مرحله اول مساله بهینه سازی با محدودیت باید به مساله بهینه

سازی بدون محدودیت معادلش تبدیل شود. در مرحله دوم بر اساس مساله بهینه سازی بدون محدودیت یک سیستم دینامیک معادلات دیفرانسیل طراحی شود و در مرحله آخر بر اساس سیستم دینامیک مدار VLSI آن طراحی گردد. مساله GAP را به صورت زیر به یک مساله برنامه ریزی غیر خطی تبدیل می کنیم:

$$Min \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}$$

s.t.

$$\sum_{j=1}^{n} a_{ij}x_{ij} \le b_i, \qquad i \in I = \{1,2,\cdots,m\},$$

$$\sum_{i=1}^{m} x_{ij} = 1, \qquad j \in J = \{1,2,\cdots,n\},$$

$$x_{ij}(1-x_{ij}) = 0, \qquad \forall i \in I, j \in J.$$

حال با استفاده از روش تابع جریمه این مساله را به یک مساله بدون محدودیت تبدیل می کنیم.

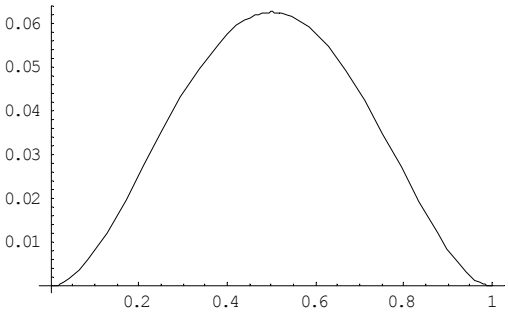
$$E(x,k) = \sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} + \frac{k}{2}\sum_{j=1}^{n}\left(\sum_{i=1}^{m} x_{ij} - 1\right)^2 + \frac{k}{2}\sum_{i=1}^{m}\min^2\left\{0, b_i - \sum_{j=1}^{n} a_{ij}x_{ij}\right\} + \frac{k}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}^2(1-x_{ij})^2$$

که در آن k پارامتر جریمه است.

برای این تابع هدف بدون محدودیت سیستم دینامیک معادلات دیفرانسیل آن را به صورت زیر تشکیل می دهیم:

$$\frac{dx_{pq}}{dt} = -\mu\left[c_{pq} + k\left(\sum_{i=1}^{m} x_{iq} - 1\right) - ka_{pq}\min\left\{0, b_p - \sum_{j=1}^{n} a_{pj}x_{pj}\right\} + kx_{pq}(1-x_{pq})(1-2x_{pq})\right]$$

که در آن μ نرخ یادگیری است و $p = 1,2,\ldots m$, $q = 1,2,\ldots,n$.



شکل ۱: نمودار جمله جریمه $x_{ij}^2(1-x_{ij})^2$

همان گونه که در شکل ۱ ملاحظه می شود وقتی سیستم دینامیک از یک نقطه شدنی به نقطه دیگر حرکت می کند جمله جریمه $\frac{k}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}x_{ij}^{2}(1-x_{ij})^{2}$ دارای کران بالای حداقل $\frac{k}{32}$ است. اگر پارامتر جریمه کوچک انتخاب شود محدودیت های مساله از جمله محدودیت های صفر و یک بودن متغیرها برقرارنمی شوند و اگر بزرگ انتخاب گردد آنگاه سیستم دینامیک نمی تواند از یک جواب شدنی به جواب شدنی دیگر حرکت کند و به یک نقطه می چسبد. چون روش تابع جریمه نیاز به پارامتر جریمه بزرگ برای برقراری محدودیت های صفر و یک دارد که منجر به همگرایی سیستم به یک جواب شدنی شود از روش ضرایب لاگرانژین افزوده برای تشکیل سیستم دینامیک استفاده می کنیم. ضرایب لاگرانژین بر اساس شدنی بودن نقطه تغییر می کنند بنابراین نقطه تعادل نهایی می تواند شدنی باشد و سیستم دینامیک در مراحل اولیه به یک جواب شدنی نمی چسبد.

حال از روش ضرایب لاگرانژین افزوده استفاده می کنیم و مساله بهینه سازی دارای محدودیت را به مساله یافتن نقطه زینی تابع لاگرانژین افزوده تبدیل می نماییم. تابع لاگرانژین افزوده را با به کار بردن جریمه برای آزاد کردن محدودیت های تساوی و نامساوی و ضرایب لاگرانژین برای آزاد سازی محدودیت های صفر و یک تعریف می کنیم.

$$L(x,k,\lambda)=\sum_{i=1}^{m}\sum_{j=1}^{n}c_{ij}x_{ij}+\frac{k}{2}\sum_{j=1}^{n}\left(\sum_{i=1}^{m}x_{ij}-1\right)^{2}+\frac{k}{2}\sum_{i=1}^{m}\min^{2}\left\{0,b_{i}-\sum_{j=1}^{n}a_{ij}x_{ij}\right\}+\sum_{i=1}^{m}\sum_{j=1}^{n}\lambda_{ij}x_{ij}\left(1-x_{ij}\right)$$

که در آن $\lambda=[\lambda_{11},\ldots,\lambda_{1n},\ldots,\ldots,\lambda_{m1},\ldots,\lambda_{mn}]$ بردار ضرایب لاگرانژ است. حال سیستم دینامیک زیر را در نظر می گیریم که نقطه تعادل آن نقطه زینی تابع لاگرانژین افزوده است.

$$\frac{dx_{pq}}{dt}=-\mu\left[c_{pq}+k\left(\sum_{i=1}^{m}x_{iq}-1\right)-ka_{pq}\min\left\{0,b_{p}-\sum_{j=1}^{n}a_{pj}x_{pj}\right\}+\lambda_{pq}\left(1-2x_{pq}\right)\right]$$

$$\frac{d\lambda_{pq}}{dt}=\mu x_{pq}(1-x_{pq})$$

که در آن μ نرخ یادگیری است و $q=1،2،\ldots،n$; $p=1،2،\ldots،m$

چون در روش ضرایب لاگرانژین افزوده ضرایب لاگرانژ در جهت نقض محدودیت های صفر و یک تغییر می کنند بنابراین سیستم در مراحل اولیه به یک جواب شدنی صفر و یک نمی چسبد و می تواند به جواب بهین یا نزدیک بهین برسد.

٤ مثال عددی

از روش های گفته شده برای حل مثال زیر استفاده می کنیم.

مثال٤-١: سه مرکز سرویس و هشت مصرف کننده وجود دارند که باید سرویس داده شوند ماتریس ضرایب هزینه به صورت زیر است:

$$C=\left[c_{ij}\right]_{3\times 8}=\begin{bmatrix}1.2 & 2.6 & 4.7 & 5.8 & 0.6 & 3.2 & 3.2 & 12.8\\ 8.4 & 6.3 & 1.4 & 2.5 & 3.2 & 4.2 & 8.5 & 2.4\\ 4.2 & 2.8 & 5.2 & 2.0 & 1.6 & 1.2 & 2.4 & 8.2\end{bmatrix}$$

ماتریس A ماتریس تقاضاهای هشت مصرف کننده در زیر داده شده.

$$A=\left[a_{ij}\right]_{1\times 8}=\begin{bmatrix}4.2 & 2.6 & 6.8 & 3.5 & 4.8 & 8.4 & 1.6 & 5.4\end{bmatrix}$$

ظرفیت های سرویس سه مرکز سرویس دهی به صورت ذیل می باشد:

$$B=\left[b_{i}\right]_{1\times 3}=\begin{bmatrix}14.0 & 12.0 & 16.0\end{bmatrix}$$

با استفاده از روش شاخه و کران مقدار هدف بهین و جواب بهین محاسبه می شود که این مقادیر در جدول ۱ داده شده است. f^* نشان دهنده مقدار هدف بهین می باشد. ابتدا از روش تابع جریمه استفاده نموده و سیستم معادلات دیفرانسیل را تشکیل می دهیم. از روش رونگه کوتای مرتبه ۴ برای شبیه سازی شبکه عصبی و حل سیستم دینامیک معادلات دیفرانسیل استفاده می کنیم. با سه مقدار اولیه متفاوت که در جداول ۳،۲ و۴ داده شده اند حل را آغاز می نماییم.

	$x_{1.}^*$	$x_{2.}^*$	$x_{3.}^*$	$\sum_{i=1}^{3} x_{i.}^*$
$x_{.1}^*$	1	0	0	1
$x_{.2}^*$	1	0	0	1
$x_{.3}^*$	1	0	0	1
$x_{.4}^*$	0	1	0	1
$x_{.5}^*$	0	0	1	1
$x_{.6}^*$	0	0	1	1
$x_{.7}^*$	0	0	1	1
$x_{.8}^*$	0	1	0	1
$\sum_{j=1}^{8} a_{.j}^* x_{.j}^*$	13.6	8.9	14.8	$f^*=18.6$

جدول ۱: جواب بهین و مقدار هدف بهین

	$x_{1.}^{A0}$	$x_{2.}^{A0}$	$x_{3.}^{A0}$	$\sum_{i=1}^{3} x_{i.}^{A0}$
$x_{.1}^{A0}$	0	0	0	0
$x_{.2}^{A0}$	0	0	0	0
$x_{.3}^{A0}$	0	0	0	0
$x_{.4}^{A0}$	0	0	0	0
$x_{.5}^{A0}$	0	0	0	0
$x_{.6}^{A0}$	0	0	0	0
$x_{.7}^{A0}$	0	0	0	0
$x_{.8}^{A0}$	0	0	0	0
$\sum_{j=1}^{8} a_{.j} x_{.j}^{A0}$	0.0	0.0	0.0	$f^{A0}=0.0$

جدول ۲: جواب اولیه A

	$x_{1.}^{B0}$	$x_{2.}^{B0}$	$x_{3.}^{B0}$	$\sum_{i=1}^{3} x_{i.}^{B0}$
$x_{.1}^{B0}$	0.2	0.7	0.1	1.0
$x_{.2}^{B0}$	0.3	1.24	0.14	1.68
$x_{.3}^{B0}$	1.2	0.24	0.4	1.84
$x_{.4}^{B0}$	1.16	0.16	0.16	1.48
$x_{.5}^{B0}$	0.14	0.8	0.24	1.18
$x_{.6}^{B0}$	1.32	0.24	0.32	1.88
$x_{.7}^{B0}$	0.28	0.76	0.16	1.2
$x_{.8}^{B0}$	1.24	0.22	0.12	1.58
$\sum_{j=1}^{8} a_{.j} x_{.j}^{B0}$	20.64	15.66	27.41	$f^{B0}=58.41$

جدول۳: جواب اولیه B

	$x_{1.}^{C0}$	$x_{2.}^{C0}$	$x_{3.}^{C0}$	$\sum_{i=1}^{3} x_{i.}^{C0}$
$x_{.1}^{C0}$	0.2	0.17	1.1	1.47
$x_{.2}^{C0}$	0.3	0.24	0.84	1.38
$x_{.3}^{C0}$	0.2	0.14	1.4	1.74
$x_{.4}^{C0}$	0.16	1.16	0.16	1.48
$x_{.5}^{C0}$	0.14	0.8	0.24	1.18
$x_{.6}^{C0}$	1.32	0.24	0.32	1.88
$x_{.7}^{C0}$	0.28	0.76	0.16	1.2
$x_{.8}^{C0}$	1.24	0.22	0.12	1.58
$\sum_{j=1}^{8} a_{.j} x_{.j}^{C0}$	22.44	14.61	21.63	$f^{C0}=57.26$

جدول٤: جواب اولیه C

	$x_{1.}^{A^*}$	$x_{2.}^{A^*}$	$x_{3.}^{A^*}$	$\sum_{i=1}^{3} x_{i.}^{A^*}$
$x_{.1}^{A^*}$	1.05	-0.01	0.07	1.11
$x_{.2}^{A^*}$	1.05	-0.01	0.07	1.11
$x_{.3}^{A^*}$	0.06	0.99	0.06	1.11
$x_{.4}^{A^*}$	0.06	0.0	1.05	1.11
$x_{.5}^{A^*}$	1.05	0.0	0.07	1.11
$x_{.6}^{A^*}$	0.07	-0.01	1.05	1.12
$x_{.7}^{A^*}$	0.07	-0.01	1.05	1.11
$x_{.8}^{A^*}$	0.04	1.0	0.06	1.1
$\sum_{j=1}^{8} a_{.j} x_{.j}^{A^*}$	13.71	11.96	21.63	$f^{A^*}=17.0$

جدول ۵: جواب حاصل از مقدار اولیه A به روش تابع جریمه
(μ =2.0, K=10.0, step-size=0.001, steps=600)

	$x_{1.}^{A^*}$	$x_{2.}^{A^*}$	$x_{3.}^{A^*}$	$\sum_{i=1}^{3} x_{i.}^{A^*}$
$x_{.1}^{A^*}$	1.01	0.0	0.01	1.02
$x_{.2}^{A^*}$	1.02	0.0	0.02	1.04
$x_{.3}^{A^*}$	0.01	1.01	0.0	1.02
$x_{.4}^{A^*}$	0.01	0.0	1.0	1.01
$x_{.5}^{A^*}$	1.0	0.0	0.0	1.0
$x_{.6}^{A^*}$	0.0	0.01	1.0	1.01
$x_{.7}^{A^*}$	0.01	0.0	1.01	1.02
$x_{.8}^{A^*}$	0.02	1.0	0.0	1.02
$\sum_{j=1}^{8} a_{.j} x_{.j.}^{A^*}$	11.92	12.35	13.61	$f^{A^*}=14.44$

جدول ٦: جواب حاصل از مقدار اولیه A به روش تابع جریمه
(μ =2.0, K=100.0, step-size=0.001, steps=600)

	$x_{1.}^{B^*}$	$x_{2.}^{B^*}$	$x_{3.}^{B^*}$	$\sum_{i=1}^{3} x_{i.}^{B^*}$
$x_{.1}^{B^*}$	0.02	0.99	-0.04	0.97
$x_{.2}^{B^*}$	0.01	0.99	-0.02	0.98
$x_{.3}^{B^*}$	1.0	0.02	-0.03	0.99
$x_{.4}^{B^*}$	0.99	0.01	-0.01	0.99
$x_{.5}^{B^*}$	0.02	0.01	0.95	0.98
$x_{.6}^{B^*}$	0.03	0.02	0.98	1.03
$x_{.7}^{B^*}$	1.0	-0.01	-0.01	0.98
$x_{.8}^{B^*}$	-0.01	0.04	0.96	0.99
$\sum_{j=1}^{8} a_{.j} x_{.j}^{B^*}$	12.27	7.32	17.35	$f^{B^*}=38.55$

(μ =2.0, K=100.0, step-size=0.001, steps=600)

جدول ۷: جواب حاصل از مقدار اولیه B به روش

	$x_{1.}^{C^*}$	$x_{2.}^{C^*}$	$x_{3.}^{C^*}$	$\sum_{i=1}^{3} x_{i.}^{C^*}$
$x_{.1}^{C^*}$	0.01	-0.02	0.99	0.99
$x_{.2}^{C^*}$	0.0	-0.01	1.0	0.99
$x_{.3}^{C^*}$	0.0	0.0	0.99	0.99
$x_{.4}^{C^*}$	0.0	1.0	0.0	1.0
$x_{.5}^{C^*}$	0.0	0.99	0.0	0.99
$x_{.6}^{C^*}$	0.99	-0.01	0.0	0.98
$x_{.7}^{C^*}$	0.0	0.98	0.0	0.98
$x_{.8}^{C^*}$	0.96	0.01	0.0	0.97
$\sum_{j=1}^{8} a_{.j} x_{.j}^{C^*}$	13.54	9.75	13.49	$f^{C^*}=41.34$

جدول ۸: جواب حاصل از مقدار اولیه C به روش تابع جریمه

(μ =2.0, K=100.0, step-size=0.001, steps=600)

	$x_{1.}^{A^*}$	$x_{2.}^{A^*}$	$x_{3.}^{A^*}$	$\sum_{i=1}^{3} x_{i.}^{A^*}$
$x_{.1}^{A^*}$	0.98	0.01	0.0	0.99
$x_{.2}^{A^*}$	1.01	0.0	0.01	1.02
$x_{.3}^{A^*}$	0.99	0.01	0.0	1.0
$x_{.4}^{A^*}$	0.01	1.0	0.0	1.01
$x_{.5}^{A^*}$	0.01	1.0	0.0	1.01
$x_{.6}^{A^*}$	0.01	0.0	1.01	1.02
$x_{.7}^{A^*}$	0.0	0.01	1.0	1.01
$x_{.8}^{A^*}$	0.01	1.01	0.0	1.01
$\sum_{j=1}^{8} a_{.j} x_{.j}^{A^*}$	13.70	9.08	14.92	$f^{A^*}=19.09$

جدول ۹: جواب حاصل از مقدار اولیه A به روش ضرایب لاگرانژین افزوده

(μ =2.0, K=100.0, step-size=0.001, steps=1200)

	$x_{1.}^{B^*}$	$x_{2.}^{B^*}$	$x_{3.}^{B^*}$	$\sum_{i=1}^{3} x_{i.}^{B^*}$
$x_{.1}^{B^*}$	0.99	0.0	0.01	1.0
$x_{.2}^{B^*}$	0.99	0.01	0.0	1.0
$x_{.3}^{B^*}$	1.01	0.0	0.01	1.02
$x_{.4}^{B^*}$	0.0	0.99	0.01	1.0
$x_{.5}^{B^*}$	0.01	0.0	1.0	1.01
$x_{.6}^{B^*}$	0.0	0.01	1.0	1.01
$x_{.7}^{B^*}$	0.01	0.0	1.01	1.02
$x_{.8}^{B^*}$	0.0	0.99	0.01	1.0
$\sum_{j=1}^{8} a_{.j} x_{.j}^{B^*}$	13.66	8.92	15.06	$f^{B^*}=18.94$

جدول ۱۰: جواب حاصل از مقدار اولیه B به روش ضرایب لاگرانژین افزوده

	$x_{1.}^{C*}$	$x_{2.}^{C*}$	$x_{3.}^{C*}$	$\sum_{i=1}^{3} x_{i.}^{C*}$
$x_{.1}^{C*}$	0.99	0.01	0.0	1.0
$x_{.2}^{C*}$	0.0	0.01	1.0	1.01
$x_{.3}^{C*}$	0.98	0.0	0.01	0.99
$x_{.4}^{C*}$	0.01	0.0	0.99	1.0
$x_{.5}^{C*}$	0.0	0.99	0.01	1.0
$x_{.6}^{C*}$	0.02	0.0	0.99	1.01
$x_{.7}^{C*}$	1.01	0.01	0.0	1.02
$x_{.8}^{C*}$	0.01	0.98	0.01	1.0
$\sum_{j=1}^{8} a_{.j} x_{.j}^{C*}$	12.67	10.13	14.55	$f^{C*}=21.15$

جدول ۱۱: جواب حاصل از مقدار اولیه C به روش ضرایب لاگرانژین افزوده

(μ =2.0, K=100.0, step-size=0.001, steps=1200)

نقاط تعادل به دست آمده با استفاده از روش تابع جریمه و جواب های اولیه بالا در جداول ۵، ۶، ۷ و ۸ داده شده. همان گونه که از جدول ۵ ملاحظه می شود اگر پارامتر جریمه کوچک در نظر گرفته شود انگاه نقطه تعادل از ناحیه شدنی دور است. از طرف دیگر اگر پارامتر جریمه بزرگ در نظر گرفته شود آنگاه سیستم دینامیک به نزدیکترین نقطه شدنی می چسبد. این مساله از جداول ۶و۷و۸ ملاحظه می شود.

همان گونه که در جداول ۹، ۱۰و ۱۱ نشان داده شده شبکه عصبی لاگرانژین افزوده نتایج بهتری نسبت به شبکه عصبی جریمه می دهد. اگر با جواب اولیه A یا B آغاز کنیم به جواب بهین می رسیم جواب حاصل از C نزدیک بهین نیست اما یک جواب شدنی است که هنوز جواب شدنی بهتری می باشد.

۵ نتیجه

مساله GAP تعمیم مساله AP است که یک مساله NP-hard می باشد. در این مقاله روش شبکه عصبی را برای حل این مساله به کار می بریم و دو نوع ساختار یکی بر اساس روش تابع جریمه و دیگری ضرایب لاگرانژین افزوده را به کار برده و با هم مقایسه می کنیم. ملاحظه می شود که روش شبکه عصبی بر مبنای تابع جریمه نتایج خوبی نمی دهد در حالی که روش ضرایب لاگرانژین افزوده تا حدی این مشکل را بر طرف می کند.

منابع

[1] J. Bruck and J. W. Goodman, 1988. "A generalized Convergence Theorem for Neural Networks and its Application sin Optimization", *IEEE Transaction on Informations Theory* Vol. 34, pp. 1089-1092.

[2] R. E. Burkard, 1984."Quadratic Assignment Problems", *European Journal of Operational Research,* 15, 283-289.

[3] A. Cichocki, and R. Unbehauen, 1993. *Neural Networks for Optimization and Signal Processing,* Wiley & Sons, Chichester.

[4] L. Cooper, 1963. "Location-allocation Problems", *Operations Research*, Vol. 11, No. 3, pp. 331-344.

[5] D. Gong, M. Gen, G. Yamazaki and W. Xu, 1995. Neural Network Approach for General Assignment Problem. *Proceedings International Conference on Neural Networks* 4, Perth, 1861-1866.
[6] J. Hopfield, and D. W. Thank, 1985. "Neural computation of decisions in optimization problems", *Biological Cybernet,* Vol. 52 pp.141-152.
[7] C. Looi, 1992. "Neural Network Methods in Combinatorial Optimization", *Computers and Operations Research,* Vol. 19, No.3/4, pp.191-208.
[8] S. Matsuda, 1994. "Theoretical Characterizations of Possibilities and Impossibilities of Hopfield Neural Networks in Solving Combinatorial Optimization Problems", *Proc. Of IEEE International Conference on Neural Networks,* pp. 4563-4566.
[9] Katta G. Murty, 1992. *Network Programming,* Prentice-Hall, Inc., New Jersey.
[10] H. Taha, 1975. *Integer programming:Theory, Applications and computations,* Academy Press, New York.