ارایه یک شبکه عصبی ترکیبی برای حل مسأله فروشنده دوره‌گرد(TSP)

مهدی سعادتمند، محمد رضا اکبرزاده توتونچی، مرتضی خادمی
دانشکده مهندسی دانشگاه فردوسی مشهد، ایران
P.O. Box: 91775-1111, Mashhad, Iran
m_saadatmand_tarzjan@hotmail.com

کلید واژه‌ها
Traveling Salesman Problem (TSP)، شبکه‌های عصبی رقابتی، شبکه عصبی کوهونن، شبکه عصبی هاپفیلد.

خلاصه

شبکه عصبی هاپفیلد به خاطر ساختار ساده و فیدبکی‌اش، در حل مسایل بهینه‌سازی از جمله TSP مورد توجه است. از طرفی شبکه عصبی کوهونن به دلیل شیوه آموزش رقابتی و خصوصیات مکانشناسی‌اش(topological)، ابزاری مناسب برای حل این مسأله می‌باشد. هر کدام از این دو شبکه در کنار مزیتهایشان دارای معایبی نیز هستند. پاسخهای هاپفیلد به علت حضور مینیمم‌های محلی در تابع انرژی‌اش چندان قابل قبول نیست. سرعت کوهونن نیز در پاسخ به این مسأله کم است و از آن نمی‌توان برای کاربردهای بلادرنگ استفاده کرد. در این مقاله یک شبکه فیدبکی-رقابتی ارائه شده است که مانند کوهونن پاسخهای قابل قبولی به TSP می‌دهد و مانند هاپفیلد سریع است. شبیه‌سازی‌های انجام شده دقت و سرعت شبکه را برای TSPهای ۳۰ شهری در مقایسه با شبکه عصبی کوهونن و برایTSPهای ۱۰۰ شهری در مقایسه با چند آلگوریتم دیگر نشان می‌دهد. همچنین پاسخ شبکه به یک TSPی ۲۸۰ شهری نیز ارائه شده است.

۱- مقدمه

مسایل بهینه‌سازی ترکیبی، از قبیل مسأله فروشنده دوره‌گرد (TSP) از خانواده مسایل NP-Complete هستند [2]. با پیشرفت تکنولوژی، نیاز به حل سریعتر و مناسبتر این مسایل افزایش یافته است. تعیین مسیر حرکت مته برای سوراخ‌کردن صفحه‌های PCB [7]، تعیین مسیر بهینه انتقال داده در شبکه‌های کامپیوتری [3]، پردازش تصویر و تشخیص الگو از جمله زمینه‌هایی هستندکه حل TSP برایشان بسیار راه‌گشاست. TSP، همانطور که از نامش برمی‌آید، عبارت است از یافتن کوتاهترین مسیر بستهٔ ممکن بین N شهر. هدف ما در این مقاله، حل بلادرنگ TSP به کمک شبکه‌های عصبی است. در این مقاله شبکه عصبی جدیدی ارائه شده که می‌تواند به این مسأله در زمان کوتاهی پاسخهای چشمگیری دهد.

تا کنون شبکه‌های عصبی متعددی برای حل TSP ارایه شده است از آنجمله می‌توان از شبکه‌های کوهونن [5] ، هاپفیلد [8] و Boolean [9] نام برد. در این مقاله تنها دو شبکهٔ کوهونن و هاپفیلد مورد توجه ماست. مدل هاپفیلد دارای دو مشخصه اساسی است، اول اینکه دارای تابع انرژی‌ای است که مشخصات فیزیکی شبکه و رفتار آن را بیان می کند و در ثانی به صورت یک فیدبک منفی عمل کرده و در طول آموزش، تابع انرژی سیر نزولی دارد، در حالی که مشخصه اساسی مدل کوهونن روش آموزش رقابتی آن است. هر کدام از مدلهای کوهونن و هاپفیلد دارای مزایا و معایبی هستند. با این که سرعت همگرایی و دقت کوهونن به مقدار پارامترهایش بستگی دارد ولی در مجموع می توان گفت که سرعتش کم است و به همین دلیل نمی‌توان از آن در کاربردهای بلادرنگ استفاده نمود. در مقایسه، مدل هاپفیلد از سرعت قابل قبولی برخوردار است و از این جهت برای استفادهٔ بلادرنگ مناسب است ولی اشکال اساسیِ آن وجود مینیممهای محلی در تابع انرژی‌اش است. متأسفانه این مشکل در شبکه‌هایی که تا کنون برای حل TSP بر اساس این مدل ارایه شده‌اند، حادتر هم می‌باشد. زیرا همانطور که در بخش دو توضیح داده خواهد شد، با این که TSP ذاتاً مساله‌ای خطی است، ولی تابع انرژی این شبکه‌ها عموماً مجموعی از چندجمله‌ای‌های

درجه دو است(معادله (۷)). پس مسأله‌ای خطی توسط یک تابع درجه دو مدل شده است [4]. به همین دلیل پاسخهای شبکه هاپفیلد چندان معتبر نیست[8]. همچنین در هر جمله تابع انرژی، ثابتی ضرب می‌شود که بیانگر اهمیت آن جمله نسبت به دیگر جملات تابع انرژی است(ثوابت A،B،CوD در معادله (۷)). این ایده به ظاهر ساده، منشأ اشکال اساسی دیگری در مدل هاپفیلد است و آن حساسیت زیاد شبکه به پارامترهای تابع انرژی‌اش است که با افزایش تعداد شهرهای مسیر بیشتر هم می‌شود [8].

تا کنون تلاشهای زیادی جهت فرار از مینیممهای محلی و یا حذف آنها از تابع انرژی انجام شده است. مثلا در [10] سعی شده است که مینیممهای محلی تابع انرژی، با اضافه کردن یک ورودی تصحیح کننده و قابل تطبیق به نرونها، حذف شوند. ولی علی رغم بهبودهای قابل توجه در این زمینه شبکه‌ها هنوز هم به شدت به پارامترهای تابع انرژی حساس هستند در حقیقت اکثر تلاشهایی که تاکنون در این راه انجام شده است به پیچیده‌تر شدن تابع انرژی و شبکه، منجر شده است، در حالی که یک راه مناسب برای کم کردن اثر مینیممهای محلی، ساده‌تر کردن تابع انرژی شبکه عصبی است. در این مقاله سعی شده است با تلفیق این دو ایده و ارایه یک شبکه عصبی جدید از قابلیتهای این دو شبکه به صورت توأم استفاده شود، بنابراین شبکه عصبی پیشنهادی مانند کوهونن قابل اعتماد و مانند هاپفیلد سریع است و در عین حال نیاز به تنظیم پارامترهای تابع انرژی نیز ندارد.

در بخش بعدی این مقاله تابع انرژی شبکه معرفی می‌گردد. در بخش سه ساختارش توضیح داده شده است. در بخش چهار نحوه آموزش شبکه بیان شده و همگرایی شبکه اثبات گردیده است. در بخش پنج نتایج تجربی بدست آمده از شبیه‌سازی شبکه ارائه شده است. و در بخش آخر جمع‌بندی مقاله ارائه گردیده است.

۲- تابع انرژی شبکه عصبی پیشنهادی

درتمام این مقاله موارد ذیل در نمایش کمیتها رعایت شده‌اند:

۱- درایه واقع در سطر i ام و ستون jام از ماتریس V را با $v_{i,j}$ نشان می‌دهیم.

۲- N تعداد شهرها، $d_{i,j}$ فاصله شهر i ام تا شهر jام و D ماتریس فاصله شهرها از یکدیگر است. هدف ما در این مقاله حل مسأله فروشنده دوره‌گرد متقارن(Symmetric TSP) است. در STSP فاصله شهر i ام تا شهر jام برابر با فاصله شهر jام تا شهر i ام است. بدیهی است که به‌این‌ترتیب D یک ماتریس متقارن خواهد بود.

$$d_{i,j} = \text{Distance between ith \& jth city} \qquad i = 1,2,\cdots,N \;,\quad j = 1,2,\cdots,N$$
$$D = \left[d_{i,j} \right]_{N \times N} \qquad d_{i,j} = d_{j,i} \qquad (۱)$$

۳- بردارها و ماتریسها با حروف بزرگ و درایه‌های آنها با حروف کوچک نشان داده می‌شوند. علاوه براین بردارها با علامت بردار مشخص می‌شوند، مانند: $\vec{X}$.

۴- عبارت x_k^l ، درایه k ام از بردار خروجی نرون i ام که در لایه l ام شبکه واقع است را نشان می‌دهد. همچنین نرون صفر، همان نرون N ام و نرون $N+1$ ام همان نرون اول است.

TSP را می‌توان مطابق معادله‌های زیر به صورت یک تابع خطی مدل کرد [4].

$$v_{i,j} = \begin{cases} 1 & \text{If ith city is located immediately after jth city} \\ 0 & \text{Otherwise} \end{cases} \qquad (۲)$$

$$L = \sum_{i=1}^{N} \sum_{j=1}^{N} d_{i,j} v_{i,j} \qquad (۳)$$

که L طول مسیر است. برای اینکه مسیر از هر شهر فقط یکبار بگذرد، باید شرایط ذیل نیز برقرار باشد.

$$\sum_{i=1}^{N} v_{i,j} = 1 \qquad \forall j = 1,2,\ldots,N \qquad (۵)$$

$$\sum_{j=1}^{N} v_{i,j} = 1 \qquad \forall i = 1,2,\ldots,N \qquad (۶)$$

طبق روش رایج در حل TSP با استفاده از مدل کوهونن، برای هر شهر از مسیر یک ،یا به طور کلی‌تر، تعداد ثابتی نرون در نظر گرفته می‌شود و در طول آموزش، وزنهای هر نرون به سمت مختصات شهری که باید جای آن را در مسیر بگیرد، همگرا می‌شود [5]. در حل TSP به کمک شبکه‌هایی که بر اساس ساختار هاپفیلد طراحی شده‌اند، برای هر شهر، N نرون در نظر گرفته می‌شود و خروجی نرون jام از نرونهای وابسته به شهر i ام زمانی برابر یک است که شهر i ام در مسیر در محل jام قرار داشته باشد [3][8]. تابع انرژی شبکه نیز برابر با حاصل جمع چندین چندجمله‌ای درجه دو است که هر کدام از آنها وظیفه شبیه‌سازی یکی از معادله‌های (۴) تا (۶) را بر عهده دارد. معادله زیر، تابع انرژی شبکه‌ای که Hopfield و Tank برای حل TSP پیشنهاد کردند را نشان می‌دهد [8].

$$E = \frac{A}{2}\sum_{x=1}^{N}\sum_{i=1}^{N}\sum_{\substack{j=1\\ j\neq i}}^{N} v_{x,i}v_{x,j} + \frac{B}{2}\sum_{i=1}^{N}\sum_{x=1}^{N}\sum_{\substack{y=1\\ y\neq x}}^{N} v_{x,i}v_{y,i} + \frac{C}{2}\left[\sum_{x=1}^{N}\sum_{i=1}^{N} v_{x,i} - N\right]^2 + \frac{D}{2}\sum_{x=1}^{N}\sum_{\substack{y=1\\ y\neq x}}^{N}\sum_{i=1}^{N} d_{x,y}v_{x,i}(v_{y,i+1} + v_{y,i-1}) \qquad (۷)$$

در این معادله، دو جمله اول مانع ایجاد حلقه در مسیر می‌شوند (شرایط معادله‌های(۵)و(۶))، جمله سوم باعث می‌شود تا مسیر از تمام شهرها بگذرد(شرط معادله(۴))، و در نهایت جمله آخر باعث مینیمم شدن طول مسیر می‌شود (شرط معادله(۳)). همانطور که می‌بینید شبیه‌سازی معادله‌های (۴)،(۵)و(۶) در تابع انرژی به پیچیده شدن آن و افزایش شدید مینیمم‌های محلی‌اش، می‌انجامد. همچنین این کار باعث ورود پارامترهای A، B، C و D به تابع انرژی می‌شود که نیاز به مقداردهی اولیه دارند. ما می‌خواهیم تابع انرژی شبکه‌مان تنها شامل معادله (۳) باشد و سایر شرایط مسیر در الگوریتم آموزشی رعایت شود. بنابراین خروجی نرون i ام در لایه اول شبکه را چنین تعریف می‌کنیم:

$$ {}^{i}\vec{X}^{1} = \left[{}^{i}x_j^{1}\right] \qquad \forall i = 1,2,\ldots,N \qquad (۸)$$

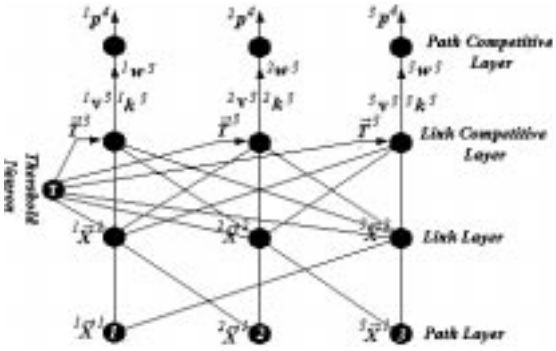
$$ {}^{i}x_j^{1} = \begin{cases} 1 & \text{If ith city is located at jth location} \\ 0 & \text{Otherwise} \end{cases} \qquad \forall j = 1,2,\ldots,N \qquad (۹)$$

به این‌ترتیب تابع انرژی شبکه نیز طبق معادله ذیل تعریف می‌شود (که برابر با طول مسیر است).

$$e = \sum_{k=1}^{N}\sum_{i=1}^{N}\sum_{j=1}^{N} d_{i,j}\ {}^{k}x_i^{1}\ {}^{k-1}x_j^{1} \qquad (۱۰)$$

اندیشه اصلی الگوریتم آموزش چنین است که ابتدا یک مسیر اولیه به شبکه معرفی می‌گردد و شبکه بر اساس آن به مسیر بهینه‌ای همگرا می‌شود، به طوری که در هر مرحله از آموزش - حتی قبل از پایدار شدن شبکه- مسیری که شبکه به ما معرفی می‌کند، معتبر باشد. شبکه باید همانند هاپفیلد به صورت یک فیدبک منفی عمل کند تا در نهایت به یکی از مینیممهای محلی تابع انرژی‌اش برسد.

۳- ساختار شبکه عصبی پیشنهاد شده



شکل ۱ : *شبکه عصبی پیشنهادی*

شبکه عصبی پیشنهادی شامل چهار لایه می‌باشد و خروجی نرونهایش به شکل بردار است. شکل ۱ این شبکه را برای سه شهر نشان می‌دهد. در شکل ورودی‌ها، خروجی‌ها و مقادیر آستانه نرونها نیز نشان داده شده‌اند. در این شبکه همانند مدل کوهونن، مسیر در **لایه اول (Path Layer)** با N نرون بیان می‌شود ولی وزنهای این نرونها به سمت مختصات شهرها میل نمی‌کند، بلکه به شهرها اجازه جابجایی در آنها داده می‌شود. معادله های (۸) و (۹) عملکرد نرونهای این لایه را بیان می‌کنند. **لایه دوم (Link Layer)** نیز شامل N نرون است و هر نرون یک کمان از مسیر را مشخص می‌کند، بنابراین داریم:

$$ {}^{i}\vec{X}^{2} = {}^{i}\vec{X}^{1} + {}^{i+1}\vec{X}^{1} \qquad (۱۱)$$

در **لایه سوم (Link Competitive Layer)** ، $N+1$ نرون داریم. یک نرون، نرون آستانه است. هر کدام از نرونهای باقیمانده نظیر یک شهر است (نرون اول نظیر شهر اول، نرون دوم نظیر شهر دوم و ...). معادله (۱۲)، عملکرد نرون آستانه و معادله‌های (۱۳)، (۱۴)، (۱۵) و (۱۶)، عملکرد سایر نرونهای این لایه را شرح می‌دهند. در این لایه در هر نرون، بین کمانهایِ مختلفِ مسیر، رقابتی شکل می‌گیرد و کمان برنده و وزن آن به عنوان خروجی نرون معرفی می‌شوند.

$$\vec{T}^{3} = \left\{t_k^{3}\right\} \quad , t_k^{3} = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} d_{i,j}\ {}^{k}x_i^{2}\ {}^{k}x_j^{2} \quad , k = 1,2,\cdots,N \qquad (۱۲)$$

$$X = \left[{}^{1}\vec{X}^{2} \quad {}^{2}\vec{X}^{2} \quad \cdots \quad {}^{N}\vec{X}^{2}\right] = \left[x_{i,j}\right] \qquad (۱۳)$$

$$\vec{D}_k = \left\{d_{k,j}\right\} \quad , \quad d_{k,j} \in D \qquad (۱۴)$$

$$ {}^{i}\vec{Z}^{3} = \vec{D}_i X - \vec{T} \quad , \quad \mathrm{i} = 1,2,\cdots,\mathrm{N} \qquad (۱۵)$$

$$\begin{cases} {}^{i}v^{3} = \min\limits_{j=1}^{N}\left({}^{i}z_j^{3}\right) \Big/ x_{i,j} \neq 1 \\ {}^{i}k^{3} = index\left({}^{i}v^{3}\right)_{\vec{z}} \end{cases} \qquad (۱۶)$$

که تابع $index(x)_{\vec{y}}$ درایهٔ عنصر x، از بردار $\vec{Y}$ را برمی‌گرداند و منظور از " / "، عبارت "به طوری که" است.

در **لایه چهارم (Path Competitive Layer)** نیز N نرون قرار دارد و همانند لایه سوم هر نرون نظیر یک شـهر اسـت (نرون اول نظیر شهر اول، نرون دوم نظیر شهر دوم، ... و نرون N ام نظیر شهر N ام). نرونها دارای وزنهـایی ($^{i}w^{4}\ ,\ i=1,2,...,N$) هستند که در پایان هر مرحله از آموزش باتوجه به مسیر جدیدی که شبکه معرفی می‌کنـد، تغییـر خواهنـد کـرد و از ایـن لحـاظ شبیه نرونهای شبکه کوهونن هستند. دراین‌باره در بخش بعدی بیشتر صحبت خواهیم کرد. معادله‌های (۱۷)، (۱۸) و (۱۹)، مبین رفتار نرونهای این لایه هستند.

$$^{i}v^{4}=^{i}v^{3}-^{i}w^{4}\quad ,\quad i=1,2,\cdots,N \tag{۱۷}$$

$$k=\begin{cases} index\left(\underset{i=1}{\overset{N}{min}}\left(^{i}v^{4}\right)\right) & \underset{i=1}{\overset{N}{min}}\left(^{i}v^{4}\right)<0 \\ 0 & Otherwise \end{cases} \tag{۱۸}$$

$$^{i}p^{4}=\begin{cases} ^{i}k^{3} & i=k \\ 0 & \text{Otherwise} \end{cases} \tag{۱۹}$$

در این لایه نرونی که کمترین وزن (k) را دارد، برنده می‌شود و خروجی‌اش، محلِ جدیدِ شـهرِ هـم‌ارزش را، در مسـیر مشخص می‌کند.

۴- آلگوریتم آموزش شبکه

الگوریتم آموزش شبکه در جدول یک آورده شده است.

جدول ۱ : *الگوریتم آموزش شبکه*

۱. مسیر معتبری را انتخاب می‌کنیم(مسیری معتبر است که درمعادله های (۴)،(۵) و (۶) صدق کند).
۲. خروجی نرونهای لایه اول را با توجه به مسیر اولیه تعیین می‌کنیم.
۳. وزنهای نرونهای لایه چهارم را طبق معادله (۲۰) تعیین می‌کنیم.

$$^{t}w^{4}=\sum_{q=1}^{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(d_{t,i}+d_{t,j}-d_{i,j}\right)\,^{q}x_{t}^{1}\,^{(q+1)}x_{i}^{1}\,^{(q-1)}x_{j}^{1} \tag{۲۰}$$

۴. اجازه می‌دهیم شبکه یک مرحله آموزش یابد.
۵. شهر نظیر نرون برنده در لایه چهارم را از محل فعلی‌اش به محل جدیدی که شبکه در خروجی این نرون بیان می‌کند منتقـل می‌کنیم. اگر خروجی نرون برنده p باشد، برنده باید در مسیر، بین دو شهری که کمان p ام مسیر جاری را تشـکیل می‌دهنـد قرار گیرد.
۶. اگر در لایه آخر نرونی برنده نشده باشد، شبکه پایدار شده است و آموزش شبکه پایان می‌یابد.
۷. مراحل ۳، ۴ ، ۵ و ۶ را تا آنجا تکرار می‌کنیم که شبکه پایدار شود.

فرض کنید:

۱- مسیری که شبکه در انتهای مرحله t ام (ابتدای مرحله $t+1$ ام) آموزش نشان می‌دهد $Ph(t)$ و انرژی شـبکه در ایـن حالت $e(t)$ است.

۲- در پایان مرحله $(t+1)$ ام، نرون k برنده شده است و خروجی آن p است.

$$p=^{k}p^{4} \tag{۲۱}$$

۳- شهر k ، در مسیر $Ph(t)$ در محل p_0 بوده است.

$$Ph_{p_0}^{t}=k \tag{۲۲}$$

طبق نامعادله $e(t+1)\le e(t)$ (اثبات در ضمیمه۱)، می‌توان گفت که در طی هر مرحله از آمـوزش، تـابع انـرژی سـیر نزولی دارد و به این ترتیب شبکه در نهایت به مسیری بهتر از مسیر اولیه(و یا همـان مسـیر در صورتـی کـه در همـان مرتبـه اول آموزش نرونی برنده نشود) همگرا می‌شود. با توجه به اینکه در هر مرحله از آموزش شبکه، شهری از یک مکان در مسیر، به مکـان دیگری منتقل می‌شود، می‌توان نتیجه گرفت که مسیر نهایی شبکه همان خصوصیات مسیر اولیه‌اش را دارد. یعنی اگر مسیرِ اولیه از تمام نقاط گذشته باشد، مسیر نهایی نیز از تمام آنها خواهد گذشت و اگر مسیر اولیه -به عنوان مثال- سه حلقـه داشـته باشـد، مسیر نهایی نیز سه حلقه خواهد داشت. پس برای حل TSP، اگر مسیر اولیهٔ شبکه معتبر باشد، مسیر نهایی آن نیز معتبر خواهـد بود.

۵-نتایج تجربی

شبکه ما در اولین مرحله آموزش نیاز به یک مسیرِ اولیهٔ معتبر دارد. در حقیقت شرایط معادله‌های (۴)،(۵)و(۶) توسـط مسیر اولیه به شبکه اعمال می‌شوند. در این تحقیق شش شیوه مختلف برای تولید مسیر اولیه، شبیه‌سازی شده است:

۱- مسیر تصادفی

مسیر اولیه به صورت تصادفی انتخاب شود.

۲- نزدیکترین همسایه

مسیر اولیه چنین بدست می‌آید که ابتدا نقطه‌ای به شکل تصادفی انتخاب می‌شود، سپس نزدیک‌ترین نقطه به آن انتخاب می‌شود و پس از آن نزدیکترین نقطه به نقطه دوم و ... . این کار را تا انجا ادامه می‌یابد که همه نقاط در مسیر قرار گیرند.

۳- پوسته محدب (Convex Hull [1])

در این الگوریتم ابتدا چهار شهر خارجی (خارجی‌ترین شهرها) تشکیل یک مسیر بسته می‌دهند. سپس در هر مرحله فاصله شهرهایی که روی مسیر نیستند تا اضلاع چندضلعی محاسبه می‌شود و شهری که کمترین فاصله را دارد به مسیر اضافه می‌شود. این عمل تا قرار گرفتن تمام شهرها در مسیر ادامه می‌یابد.

۴- پوسته‌ای براساس بزرگترین چهارضلعی (Hull Through Four Most Distant Points ، HTFMDP)

این آلگوریتم نیز شبیه آلگوریتم قبلی است با این تفاوت که چهار شهر اول، شهرهایی هستند که بیشترین فاصله ممکن را از هم دارند.

۵- تولید مسیر اولیه بر مبنای شبکه عصبی پیشنهادی (روش اول)

در بخش‌های قبلی ساختار شبکه و نحوه آموزش آن بیان شد. اکنون این شبکه را چنین تغییر می‌دهیم. فرض کنید لایه اول و دوم شبکه، m نرون($m<N$) داشته باشند (بنابراین این دو لایه یک مسیر m شهری را نشان خواهند داد)، لایه سوم شبکه نیز تنها شامل شهرهایی باشد که در مسیر قرار ندارند (N-m نرون) و لایه چهارم از آن حذف شده باشد. خروجی هر نرون لایه سوم، کمانی از مسیر را معرفی می‌کند که اگر شهر نظیر آن نرون -در مسیر- بین دو شهری که کمان را تشکیل می‌دهند قرار گیرد، مسیری با $m+1$ شهر ایجاد می‌شود به طوری که:

a- مسیر جدید علاوه بر شهرهای قبلی شامل شهر مورد نظر نیز است.

b- این شهر اگر در هر جای دیگر از مسیر قبلی قرار گیرد مسیری طولانی‌تر ایجاد خواهد شد (این مطلب نتیجه مستقیم معادله (۱۶) است).

بنابراین اگر از بین این نرونها، نرونی که کمترین فعالیت را دارد انتخاب کنیم آنگاه به مسیری m+1 شهری می‌رسیم، که طولش از تمام مسیرهای حاصل از جایگذاری شهرهای لایه سوم در مسیر قبلی، کوتاهتر است.

$$k = \min_{i=1}^{N}\left({}^{i}v^{3}\right) \qquad (۲۳)$$

اکنون کافی است که به لایه‌های اول و دوم شبکه یک نرون دیگر اضافه کنیم و شهر برنده را در محلی که در خروجی نرون نظیرش معرفی شده است (فرض کنید خروجی این نرون p باشد) قرار داده، نرون نظیر این شهر را از لایه سوم حذف کنیم. شهر برنده، به این ترتیب در مسیر قرار می‌گیرد که نرون جدیدِ لایه اول، بین دو نرونی که کمان p را تشکیل می‌دهند قرار می‌گیرد و خروجی آن طبق معادله ذیل تعیین می‌شود.

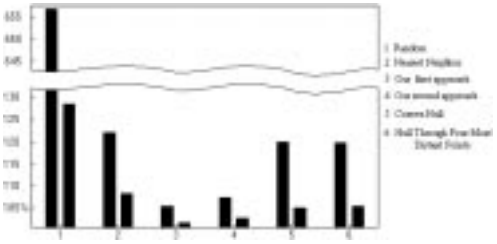
$${}^{i}x_{j}^{1} = \begin{cases} 1 & j = k \\ 0 & Otherwise \end{cases} \qquad (۲۴)$$

که k از معادله (۲۳) بدست آمده است. بنابراین می‌توانیم با شروع از مسیری با تعداد شهرهای کم به یک TSPی کامل ومعتبر برسیم و سپس آن را به شبکه بدهیم تا بهینه گردد. در اینجا برای شروع از مسیر چهار شهری‌ای که در آلگوریتم سه بیان شد، به عنوان مسیر اولیه استفاده کرده‌ایم.

۶- تولید مسیر اولیه بر مبنای شبکه عصبی پیشنهادی (روش دوم)

این روش نیز همانند بالاست، با این تفاوت که به جای مسیر چهار شهری آلگوریتم سوم، از نظیر آن در آلگوریتم چهارم، استفاده شده است.

برای نشان دادن کارایی شبکه از ۲۰۰ توزیع تصادفی ۱۰۰ شهری، به کمک شش روشی که بیان شد ۲۰۰ دسته(شش‌تایی) مسیر اولیه بدست آمد. مسیرهای اولیه هر دسته به شبکه داده شد و برای هر دسته شش مسیر بهینه بدست آمد. بنابراین برای هر توزیع، ۱۲ مسیر بدست آمد. از آنجا که طول مسیرها، وابسته به توزیع شهرهاست، لازم است که قبل از میانگین‌گیری، آنها را نرمال کرد. برای نرمال کردن، طول مسیرهای هر دسته را، بر طول کوتاهترین مسیر همان دسته(که



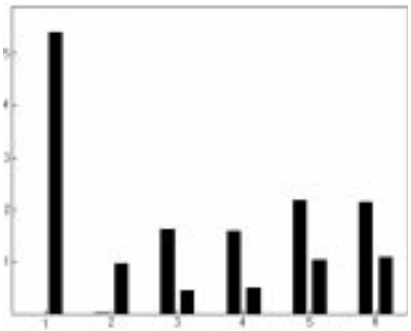
شکل ۲ : *طول متوسط مسیر اولیه (میله اول هر جفت)، مسیر نهایی شبکه (میله دوم هر جفت) نسبت به طول کوتاهترین مسیر.*

[1] الگوریتم‌های سه و چهار در [7]، به عنوان دو روش مناسب و سریع برای پاسخ به TSP پیشنهاد شده‌اند.

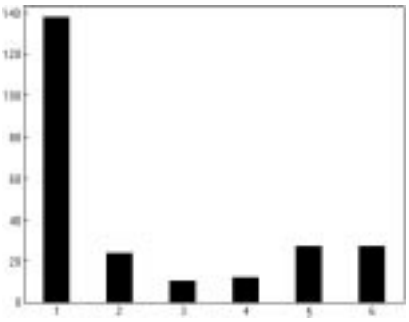
بهترین پاسخ است)، تقسیم کردیم. سپس از طولهای مسیرهای نظیر (همهٔ مسیرهای تصادفی اولیه با هم، همهٔ مسیرهای تصادفی بهینه با هم و ...) میانگین‌گرفته شد. عمل میانگین‌گیری را برای زمانهای همگرایی شبکه (در سیستم Pentium 200 تحت نرم‌افزار Matlab) و تعداد فیدبکهایش در طول مرحلهٔ آموزش (بدون نرمال کردن مقادیر) نیز انجام دادیم. نتایج به ترتیب در شکلهای ۲ ، ۳ و ۴ آورده شده است. همانطور که دیده می‌شود پاسخهای شبکه به مسیرهای اولیه‌اش وابسته است. شبکه طول مسیر تصادفی را در ۱۴۰ مرتبه فیدبک به بیش از یک پنجم کاهش داده است و این قدرت شبکه را نشان می‌دهد. بهترین مسیر اولیه و بهترین پاسخها مربوط به مسیرهایی است که از روش پنجم بدست آمده‌اند همچنین مجموع زمان ایجاد مسیر اولیه و همگرایی شبکه، در این روش از بقیه روشها کمتر است. با توجه به اینکه الگوریتم ۵ در حقیقت، از شبکه پیشنهادی منشعب شده است پس می‌توان عملیات مسیریابی شبکه را به دو فاز تقسیم نمود، تعیین مسیر اولیه و بهینه‌سازی آن. بنابراین می‌توان گفت شبکهٔ پیشنهادی، قادر است در زمان کوتاهی بدون نیاز به هیچگونه مقداردهی اولیه‌ای، یک مسیر اولیه مناسب پیشنهاد کرده، آن را بهینه کند.

از الگوریتمهای Convex Hull و HTFMDP در [7] به عنوان دو روش سریع برای حل TSP یاد شده است. همانطور که در شکل ۳ می‌توان دید، سرعت شبکه پیشنهادی، تقریبا با سرعت این دو الگوریتم برابر است در حالی که مسیرهای پیشنهادی‌اش از مسیرهای بدست آمده از این دو الگوریتم، به اندازه(بیش از) ۱۷٪ طول مسیر بهینه،کوتاهتر است.

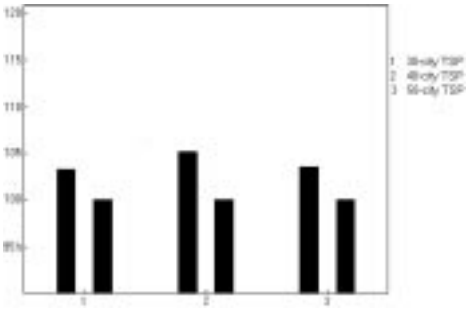
شبکه کوهونن به عنوان ابزاری با پاسخهای قابل قبول در حل TSP، شناخته شده است. به منظور تعیین دقت شبکه پیشنهادی در مقایسه با کوهونن، شبکه‌ای که در [5] ارایه شده است را شبیه‌سازی کردیم. برای ۱۰۰ توزیع ۳۰ شهری، ۱۰۰ توزیع ۴۰ شهری و ۱۰۰ توزیع ۵۰ شهری، پاسخهای هر دو شبکه و زمان همگرایی‌شان بدست آورده شد. پارامترهای شبکه کوهونن چنین انتخاب شدند $G_0 = 10$ ¥ $\eta_0 = \sqrt{30}$ که η_0 سرعت و G_0 پارامتر همسایگی آموزش را مشخص می‌کند. بنا به نظر نویسندگان [5]، شبکه با این پارامترها، بهترین پاسخها را می‌دهد. طول مسیرها را ، با روشی مشابه آنچه قبلا توضیح داده شد، نرمال کرده، از کمیتهای یکسان(تمام پاسخهای ۳۰ شهری کوهونن با هم، تمام پاسخهای ۴۰ شهری کوهونن با هم و...) میانگین گرفتیم. همچنین از زمانهای همگرایی هر دو شبکه نیز میانگین گرفته شد. نتایج در شکلهای ۵و۶ نشان داده شده است. همانطور که در شکل ۶ می‌بینید شبکه کوهونن بسیار کند است به طوری که سرعت همگرایی‌اش با شبکه پیشنهادی قابل مقایسه نیست. نمودار شکل ۵ نیز حاکی از آن است که تقریبا در تمام موارد مسیرهای شبکه پیشنهادی کوتاهتر بوده است، زیرا در شکل۵، همه میله‌های مربوط به شبکه پیشنهادی، تقریبا ۱۰۰٪ را نشان می‌دهند. مسیرهای پیشنهادی کوهونن به طور متوسط به اندازه(بیش از) ۲/۵٪ طول مسیر بهینه، بلندتر بوده‌اند.



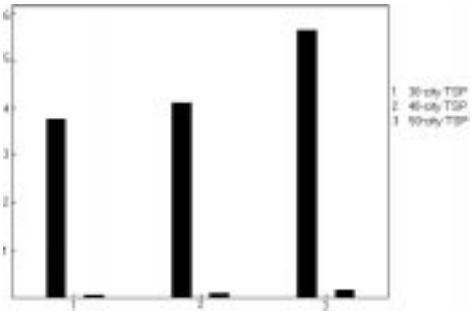
شکل ۳ : *زمان تولید مسیر اولیه (میله اول هر جفت)، زمان آموزش شبکه بر حسب ثانیه (میله دوم هر جفت).*



شکل ۴ : *تعداد فیدبکهای شبکه برای مسیرهای اولیه مختلف*

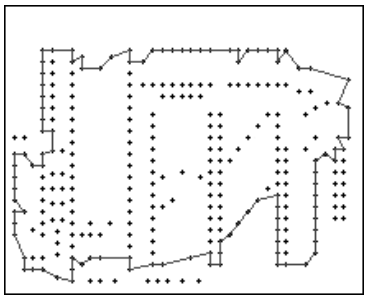


شکل ۵ : *طول متوسط مسیرهای پیشنهاد شده توسط شبکه کوهونن(میله اول هر جفت)، طول متوسط مسیرهای شبکه پیشنهادی(میله دوم هر جفت) نسبت به طول کوتاهترین مسیر.*

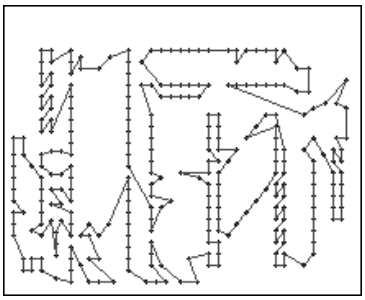


شکل ۶ : *زمان همگرایی شبکه کوهونن(میله اول هر جفت)، زمان همگرایی شبکه پیشنهادی بر حسب ثانیه (میله دوم هر جفت).*

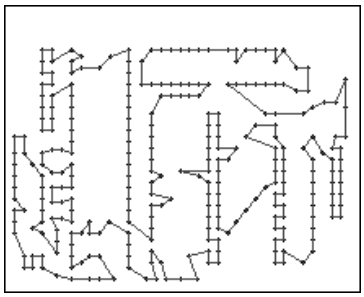
در شکل۷ مسیر حاصل از آلگوریتم ۵ پس از ۱۰۰ مرتبه تکرار، برای یک توزیع ۲۸۰ نقطه‌ای که در حقیقت نقشه یک PCB [11] است، نشان داده شده است. همچنین مسیر اولیه و نهایی شبکه پیشنهادی(برای همان توزیع) نیز در شکلهای ۸ و ۹ نشان داده شده‌اند.۰



شکل ۷ : *مسیر حاصل از آلگوریتم ۵ پس از ۱۰۰ مرتبه تکرار.*



شکل ۸ : *مسیر اولیه پیشنهاد شده با استفاده از آلگوریتم ۵.*



شکل ۹ : *پاسخ شبکه پیشنهادی به مسیر اولیه شکل ۸ .*

۶- جمع‌بندی

در این مقاله شبکه عصبی‌ای پیشنهاد شده است که باسرعت قابل قبولی پاسخهای مناسبی به مسأله فروشنده دوره‌گرد می‌دهد. شبیه‌سازیهای انجام شده دلیلی بر این مدعا است. با اینکه تابع انرژی شبکه پیشنهادی تنها شامل یک جمله است اما هنوز هم دارای مینیمم‌های محلی است و همین امر باعث حساسیت شبکه به مسیرهای اولیه‌اش شده است، گرچه که این مشکل با ارایه روشهای تولید مسیر اولیه، برمبنای شبکه پیشنهادی، به شکل چشمگیری حل شده است. از این شبکه می‌توان علاوه برکاربردهای بلادرنگ برای حل TSPهایِ بسیار بزرگ نیز استفاده کرد. قدرت این شبکه در ساختار فیدبکی-رقابتی آن است که این دو خاصیت اجازه انعطاف زیادی به شبکه می‌دهد. دیگر قابلیت مهم این شبکه، این است که می‌توان به کمک مسیر اولیه، شرایط بسیار پیچیده‌ای را در مسیر شبیه‌سازی نمود. از این ابزار می‌توان برای حل طیف وسیعی از مسایل بهینه‌سازی ترکیبی استفاده کرد.

۷- قدردانی

این مقاله حاصل تلاشی است که در گروه روباتیک دانشگاه فردوسی و با مساعدت هستهٔ علمی بسیج دانشجویی دانشکده مهندسی، آغاز شده بود. در اینجا از دست اندرکاران بسیج دانشجویی دانشکده مهندسی دانشگاه فردوسی، قدردانی و تشکر می‌شود.

۸- مراجع

[1] S. Haykin, “Neural Networks”, New York, Macmillan College Pub.,1994 .

[2] P. Crescenzi, V. Kann, “A Compendium of NP optimization problems”, http:// www.zvne.fer.hr/ ~zmija/ resources/ science_resources/ nn_for_optimization/index.html

[3] M. K. Mehmet Ali, F. Kamoun, “Neural Networks for Shortest Path Computation and Routing in Computer Networks”, IEEE Trans. Neural Networks, VOL. 4, NO. 6, Nov. 1993 .

[4] K. Smith, “An Argument for Abandoning the Traveling Salesman Problem as a Neural Network Benchmark”, IEEE Trans. Neural Networks, VOL. 7, NO. 6, Nov. 1996 .

[5] S. J. Kim, H. M. Choi, “An Efficient Algorithm for Traveling Salesman Problems based on Self-Organizing Feature Mars”, 2th IEEE Int. Conference on Fuzzy Systems, Mar. 1993.

[6] K. Shinozawa, T. Uchiyama, K. Shimohara, ”An Approach for Solving Dynamic TSPs using Neural Networks”, IJCNN, Nov. 1991 .

[7] O. Lahyani, E. Oertli, H. Eberle, “Optimizing Drill Tapes for Printed Circuit Boards”, WAC, Vol.3, May 1996 .

[8] S. M. Gowda, B. W. Lee ,B. J. Sheu, ”An Improved Neural Network Approach to the Traveling Salesman Problem”, TENCON, Nov. 1989 .

[9] S. Bhide, N. John, M. R. Kabuka, ”A Real-Time solution for the Traveling Salesman

Problem using a Boolean Neural Network", ICNN, Mar. 1993 .
[10] B. W. Lee, B. J. Sheu, "Modified Hopfield Neural Networks for Retrierving the Optimal Solution", IEEE Trans. Neural Networks, VOL. 2, NO. 1, Jan. 1991 .
[11] TSPLIB, http://softlib.rice.edu/softlib/tsplib

ضمیمه ۱

می‌خواهیم نامعادله $e(t+1) \le e(t)$ را اثبات کنیم. از (۱۶) و (۲۱)، معادله (۲۵) و از (۱۲) و (۱۵)، معادله (۲۶) نتیجه می‌شود.

$$\begin{cases} {}^{k}v^{3} = {}^{k}z_{p}^{3} \\ {}^{k}k^{3} = p \end{cases} \quad (۲۵)$$

$${}^{k}z_{p}^{3} = \sum_{j=1}^{N} d_{k,j}\ {}^{p}x_{j}^{2} - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} d_{i,j}\ {}^{p}x_{i}^{2}\ {}^{p}x_{j}^{2} \quad (۲۶)$$

از معادله (۱۱) و دو معادله قبل نتیجه می‌شود:

$${}^{k}v^{3} = \sum_{j=1}^{N} d_{k,j}\left({}^{p}x_{j}^{1} + {}^{p+1}x_{j}^{1}\right) - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} d_{i,j}\left({}^{p}x_{i}^{1} + {}^{p+1}x_{i}^{1}\right)\left({}^{p}x_{j}^{1} + {}^{p+1}x_{j}^{1}\right) \quad (۲۷)$$

$${}^{k}v^{3} = \sum_{i=1}^{N}\sum_{j=1}^{N}\left(d_{k,j} + d_{k,i}\right){}^{p}x_{j}^{1}\ {}^{p+1}x_{i}^{1} - \sum_{i=1}^{N}\sum_{j=1}^{N} d_{i,j}\ {}^{p}x_{i}^{1}\ {}^{p+1}x_{j}^{1} \quad (۲۸)$$

$${}^{k}v^{3} = \sum_{i=1}^{N}\sum_{j=1}^{N}\left(d_{k,i} + d_{k,j} - d_{i,j}\right){}^{p}x_{i}^{1}\ {}^{p+1}x_{j}^{1} \quad (۲۹)$$

می‌دانیم که شهر w در محل p_0 مسیر $Ph(t)$ است. پس از (۱۷)، (۲۰) و (۲۹) معادلات ذیل بدست می‌آیند.

$${}^{k}w^{4} = \sum_{i=1}^{N}\sum_{j=1}^{N}\left(d_{k,i} + d_{k,j} - d_{i,j}\right){}^{(p_0+1)}x_{i}^{1}\ {}^{(p_0-1)}x_{j}^{1} \qquad , \qquad {}^{p_0}x_{k}^{1} = 1 \quad (۳۰)$$

$${}^{k}v^{4} = \sum_{i=1}^{N}\sum_{j=1}^{N}\left[\left(d_{k,i} + d_{k,j} - d_{i,j}\right){}^{p}x_{i}^{1}\ {}^{p+1}x_{j}^{1}\right] - \left[\left(d_{k,i} + d_{k,j} - d_{i,j}\right){}^{(p_0+1)}x_{i}^{1}\ {}^{(p_0-1)}x_{j}^{1}\right] \quad (۳۱)$$

فرض کنید در مسیر $Ph(t)$ داشته باشیم:

$$Ph_{p_0-1}^{t} = a \quad (۳۲)$$

$$Ph_{p_0+1}^{t} = b \quad (۳۳)$$

$$Ph_{p}^{t} = c \quad (۳۴)$$

$$Ph_{p+1}^{t} = d \quad (۳۵)$$

بنابراین با توجه به (۱۸) می‌توان معادلات ذیل را نتیجه گرفت.

$${}^{k}v^{4} = \left(d_{k,c} + d_{k,d} - d_{c,d}\right) - \left(d_{k,a} + d_{k,b} - d_{a,b}\right) < 0 \quad (۳۶)$$

$$d_{k,c} + d_{k,d} + d_{a,b} < d_{k,a} + d_{k,b} + d_{c,d} \quad (۳۷)$$

انرژی شبکه در مرحلهٔ t ام آموزش برابر است با:

$$e(t) = \left[\sum_{\substack{q=1 \\ q\neq p_0,p_0+1,p+1}}^{N}\sum_{i=1}^{N}\sum_{j=1}^{N} d_{i,j}\ {}^{q}x_{i}^{1}\ {}^{(q-1)}x_{j}^{1}\right] + d_{k,a} + d_{b,k} + d_{d,c} \quad (۳۸)$$

$$e(t) = C_1 + d_{k,a} + d_{k,b} + d_{c,d} \quad (۳۹)$$

انرژی شبکه در مرحلهٔ (t+1) ام نیز به روش مشابه محاسبه می‌شود.

$$e(t+1) = C_2 + d_{k,c} + d_{k,d} + d_{a,b} \quad (۴۰)$$

تفاوت دو مسیر $Ph(t)$ و $Ph(t+1)$ در مکان شهرِ k است. k در $Ph(t)$، بین a و b قرار دارد در حالی که در $Ph(t+1)$، بین c و d قرار گرفته است. اگر از مسیر $Ph(t)$ کمانهای (a,k) ، (k,b) و (c,d) حذف شوند و کمانهای (c,k) ، (k,d) و (a,b) به آن اضافه گردند، مسیرِ $Ph(t+1)$ حاصل می‌شود. بنابراین سایر قسمتهای این دو مسیر مشترکند. C_1 و C_2 برابر با طول همین قسمتها هستند، پس با هم برابر هستند. بنابراین از معادله های (۳۷)، (۳۹) و (۴۰) می‌توان نامعادله بالا را نتیجه گرفت.