جایابی پست‌های توزیع بوسیلهٔ شبکه‌های عصبی

فردین اخلاقیان طاب — دانشگاه کردستان

جمال مشتاق — شرکت توزیع برق استان کردستان

چکیده:

در اکثر شرکت‌های توزیع، انتخاب محل مناسب پست‌های توزیع بدون استفاده از روش کلاسیک و بصورت تجربی انجام می‌گیرد که معمولاً ملاک عمل، مرکز ثقل بار و رعایت حریم الکتریکی پست می‌باشد. از آنجا که مرکز ثقل باربر اساس فاصله هوایی مصرف کننده‌ها از هم بدست می‌آید در حالیکه فاصلهٔ واقعی زمینی با توجه به وضعیت قرار گرفتن کوچه و خیابانها، غیر از فاصله هوایی می‌باشد لذا دقت عمل پایین خواهد بود. در این مقاله ضمن بررسی الگوریتم Kohonen در شبکه‌های عصبی و قابلیت آن در جایابی بهینه محل پست‌های توزیع به تعیین فاصله واقعی بارها از هم توسط برنامه‌ریزی پویا پرداخته و در نهایت نرم‌افزار تهیه شده در زمینه فوق‌الذکر و قابلیت‌های آن را مورد بررسی قرار می‌دهیم.

۱ ـ شرح مقاله:

بمنظور تغذیه مناسب مشترکین و سرمایه گذاری کمتر در احداث پست و شبکه، لازم است که پست‌های توزیع در محلهای مناسب نصب گردند. روش معمول برای این منظور، تعیین مرکز ثقل بار با استفاده از روابط ذیل می‌باشد:

$$Xp = \frac{\Sigma X_i p_i}{\Sigma p_i} \qquad (۱)$$

$$Yp = \frac{\Sigma Y_i p_i}{\Sigma p_i} \qquad (۲)$$

در روابط فوق (X_p , Y_p) مختصات محل پست و (X_i , Y_i) مختصات مشترک i ام نسبت به مبدأ دلخواه و P_i توان مصرفی مربوط به مشترک i ام است. نقایص روش فوق عبارتند از:

ـ تعیین مرکز ثقل بار، بر اساس فاصلهٔ هوایی مشترک از مبدأ صورت گرفته که این فاصله با فاصلهٔ واقعی زمینی ممکن است اختلاف قابل توجهی داشته باشد.

ـ از آنجاکه برای هر مشترک i، سه داده P_i , Y_i , X_i مورد نیاز است لذا با افزایش تعداد مشترکین حجم اطلاعات زیاد شده و پردازش داده‌ها غیر ممکن می‌گردد.

ـ اگر هدف استفاده از چند پست توزیع برای تغذیه مشترکین باشد روش فوق جوابگو نخواهد بود. با توجه به اشکالات مطرح شده، می‌توان از برنامه‌ریزی پویا برای رفع اشکال ردیف اول، چگالی بار خطی برای رفع اشکال ردیف دوم و الگوریتم شبکه‌های عصبی برای رفع اشکال ردیف سوم استفاده نمود. در این راستا، نرم‌افزاری تهیه شده که روش‌های مذکور را به هم مرتبط نموده و جایابی محلهای مناسب برای نصب پست‌های توزیع را ارائه می‌دهد.

بمنظور کاهش حجم اطلاعات بجای‌داده مربوط به تک‌تک مشترکین از چگالی بار خطی استفاده می‌نماییم به این ترتیب که با توجه به طول کوچه یا خیابان و تعداد مشترکین موجود و پیک بار همزمانی مشترکین (تجاری، مسکونی) چگالی بار خطی بر حسب آمپر بر متر بدست می‌آید. همچنین بمنظور رعایت حریم مجاز، در فایل ورودی کدبندی مناسب صورت می‌گیرد.

ورودی برنامه شامل سه ماتریس $n \times n$ است، که n تعداد نقاط تلاقی کوچه و خیابانهای مورد

نظر است. این سه ماتریس عبارتند از:

ـ ماتریس تلاقی T که اعضای آن از سه عدد تشکیل شده است. $T_{ij} = 0$ نشان دهندهٔ عدم ارتباط مستقیم دو گره i و j است و $T_{ij} = 1$ نشان دهندهٔ ارتباط مستقیم دو گره i و j بوده که در مسیر ارتباطی مشکل حریم وجود ندارد و $T_{ij} = 2$ نشان دهندهٔ ارتباط مستقیم دو گره i و j بوده که مسیر ارتباطی مشکل حریم و عدم امکان نصب ترانس وجود دارد.

ـ ماتریس مسافت DIS که فاصله گرههای دارای ارتباط مستقیم را نشان می‌دهد.

ـ ماتریس چگالی DAN که چگالی جریان هر مسیر را با توجه به دو گره انتهائی آن، مشخص می‌کند.

اینک در ذیل به بررسی الگوریتم Kohonen از شبکه‌های عصبی و برنامه‌ریزی پویا پرداخته سپس بخشهای اصلی نرم‌افزار مربوطه بیان می‌گردد.

۲ ـ شبکه‌های عصبی:

شبکه‌های عصبی به مانند مغز انسان از تعداد زیادی پردازنده کوچک (نرون) تشکیل شده است که بطور موازی اعمال بسیار ساده‌ای را انجام می‌دهند و ترکیب این عملیات‌های ساده منجر به یک محاسبه یا استنتاج گاهاً پیچیده می‌گردد. با توجه به اینکه در این شبکه امکان پردازش موازی وجود دارد سرعت این شبکه از معادل کامپیوتری خود بمراتب بالاتر است اصولاً یک شبکه عصبی با سه عامل مشخص می‌شود:

الف) مشخصهٔ نرون (رابطه ورودی ـ خروجی)

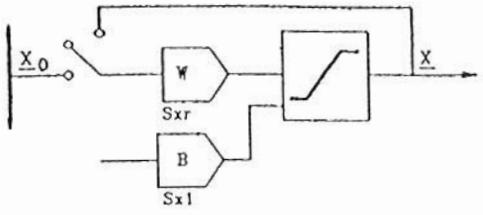
ب) شکل شبکه عصبی، که نحوه قرارگرفتن نرونها و اتصالات آنها با یکدیگر را نشان می‌دهد.

ج) قانون یادگیری شبکه، که در واقع نحوه تغییر تطبیقی وزنه‌های شبکه را نشان می‌دهد.

که در جایابی محل نصب ترانس‌ها از مشخصه خطی $Y = X$ و شبکه هاپفیلد و قانون یادگیری Kohonen استفاده شده است.

۱ـ۲ـ شبکه هاپفیلد (Hopfield Network) :

دیاگرام یک شبکه هاپفیلد بقرار ذیل است:



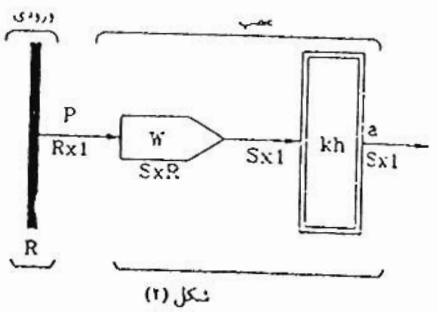
شکل (۱)

شبکه فوق ابتدا توسط تعدادی بردار X_1 ..... X_n آموزش داده می‌شود و وزنه‌های W و مقادیر آستانه θ بدست می‌آیند سپس بعد از خاتمه عمل آموزش اگر ورودی X_O به شبکه اعمال شود، بر اساس وزنه‌های W و θ و بعد از چند مرحله تکرار، خروجی X یکی از X_iها خواهد شد که بیشترین انطباق را با X_O دارد. شبکه هاپفیلد یکی از شبکه‌هایی است که در عملیات تشخیص الگو بکار می‌رود، نام دیگری که به این شبکه می‌دهند عبارتست از: **Content addresable memory**. به این معنی که اگر بخواهیم به بخشی از حافظه دسترسی پیدا کنیم و بجای آدرس آن بخش از حافظه، فقط پاره‌ای از اطلاعات آن بخش بطور ناقص در دسترس باشد می‌توان از شبکه فوق استفاده نمائیم.

۲ـ۲ـ الگوریتم Kohonen:

برای بررسی قانون یادگیری **Kohonen** ابتدا به شکل (۲) توجه می‌کنیم:

همانطور که از شکل پیداست برای عمل آموزش و تعیین وزنه‌های W هیچ فیدبکی از خروجی گرفته نمی‌شود، و در ضمن خروجی مطلوب و معلوم در این روش وجود ندارد و در حقیقت عمل آموزش بدون داشتن خروجیهای



شکل (۲)

مطلوب صورت می‌گیرد. اساساً در شبکه‌های عصبی دو نوع آموزش داریم:

۱ ـ یادگیری با واسط (superused learning) که در آن خروجیهای شبکه مشخص هستند و وزنه‌های شبکه عصبی بر اساس آنها آموزش داده می‌شوند.

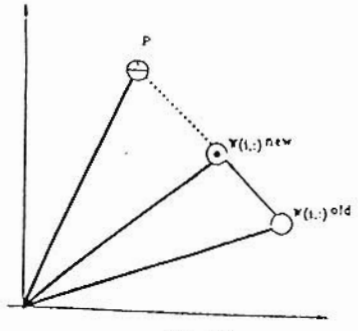
۲ ـ یادگیری بی‌واسطه (unsupervised learning) که در آن خروجیهای شبکه مشخص نیستند و وزنه‌های شبکه عصبی در مراحل تکرار مشخص، آموزش داده می‌شوند.

همانطور که توضیح داده شد الگوریتم kohonen از نوع unsupervised learning است.

الگوریتم kohonen و نحوه آموزش وزنه‌های W در شکل (۳) نشانداده شده است و فرمول ریاضی تعیین وزنه‌ها در هر مرحله تکرار عبارتست از:

$$W^{new} = (1 - \alpha)\, W^{old} + \alpha p^{T} \qquad 0<\alpha<1 \qquad (۳)$$

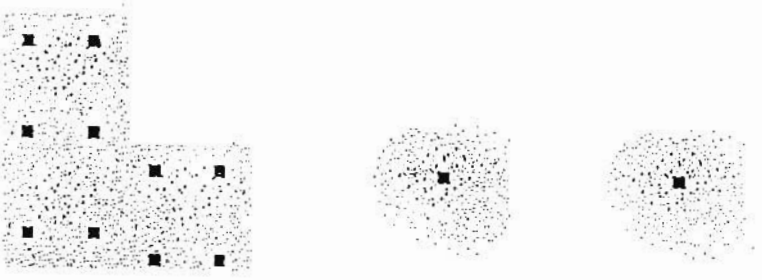
ابتدا وزنه‌های شبکه یک مقدار اولیه بخود می‌گیرند سپس با ورودی بردار p اختلاف دو بردار یعنی $P - W^{oLd}$ بدست آمده و مقدار وزنه‌های جدید باندازه $\alpha(p\text{-}w^{oLd})$ نسبت به حالت قبل تعدیل



شکل (۳)

پیدا کرده و به بردار نزدیک می‌شود. هر گاه تعداد n بردار P_1 ..... P_N داشته باشیم و از الگوریتم فوق برای آموزش وزنها استفاده کنیم می‌توان نتیجه گرفت که بعد از تعداد مراحل کافی، وزنه‌ها یک بردار خواهند شد که می‌توان بعنوان یک نماینده برای کلیه P_iها عمل کند. البته باید توجه داشت که بر حسب بردارهای ورودی P_i برای اینکه بتوان به انتخاب نماینده، کلیه اطلاعات حفظ گردد، لازم است

که تعداد مناسب نماینده برای وزنه‌ها انتخاب نمائیم. برای روشن شدن موضوع به شکلهای (۴ ـ الف) و (۴ ـ ب) توجه می‌کنیم.



شکل (۴ ـ الف)　　شکل (۴ ـ ب)

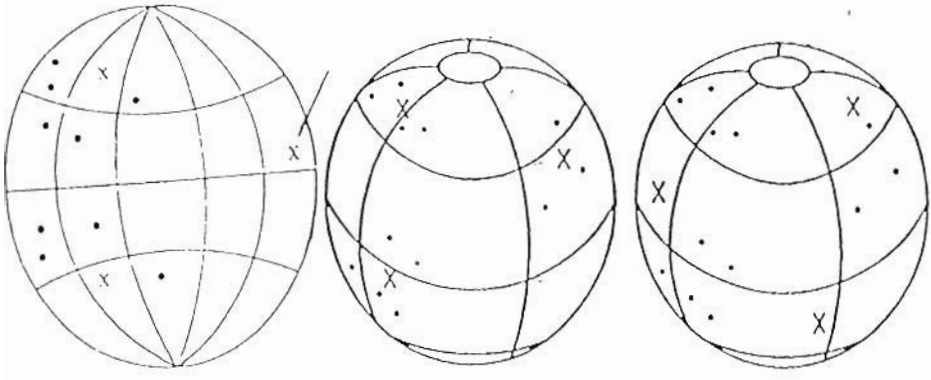
در شکل (۴ ـ الف) تعداد بی‌شماری الگو وجود دارد و با توجه به وضعیت الگوها، با انتخاب دو دسته وزنه در نهایت کار، نماینده‌ها در محل مناسب قرار می‌گیرند.

در شکل (۴ ـ ب) با توجه به وضعیت الگوها و انتخاب ده دسته وزنه، نماینده‌ها در محل مناسب خود قرار گرفته‌اند.

یکی از مسائلی که در الگوریتم **kohonen** وجود دارد، **Dead Point** است یعنی وزنه‌هایی که با توجه به وضعیت الگوها و مقادیر اولیه، هیچگاه در طول مراحل تکرار تعدیل نمی‌شوند، چراکه هر گاه تعداد نماینده‌ها بیشتر از یک باشد در هر مرحله تکرار ورودی P_i تنها با نماینده نزدیک به خود ارتباط پیدا کرده‌و طبق رابطه (۳) آن را به سمت خود نزدیک می‌کند لذا اگر مقادیر اولیه نامناسب انتخاب شوند شکل **dead point** در آموزش آشکارمی‌گردد. شکلهای (۵ ـ الف)،(۵ ـ ب)و(۵ ـ ج) موضوع فوق را به تصویر کشیده‌اند.

در شکل (۵ ـ الف) با توجه به وضعیت الگوها سه نماینده انتخاب و مقدار اولیه برای هر یک مناسب انتخاب شده است. در شکل (۵ ـ ب) وضعیت نماینده‌های انتخابی بعد از پایان آموزش نشان داده شده است و همانطور که از شکل پیداست مشکل **dead point** وجود ندارد.

شکل (۵ ـ ج) بدلیل عدم انتخاب مقدار اولیه و تعداد نماینده مناسب یکی از نمایندهها، حاوی هیچگونه اطلاعاتی نیست و در حقیقت **dead point** را نشان میدهد.



شکل (۵ ـ الف) شکل (۵ ـ ب) شکل (۵ ـ ج)

۳ـ۲ـ الگوریتم زیر برنامه شبکه عصبی:

قبل از اجرای این زیر برنامه لازم است که از برنامه ریزی پویا استفاده کرده و میزنیمم فاصله هر گره از شبکه از کل گرههای دیگر شبکه و همچنین مسیر میزنیمم را تعیین نمائیم. این قسمت از برنامه در بخش دیگر مقاله تشریح میگردد.

در ابتدا بر حسب انتخاب user، تعداد پست (تعداد نمایندههای انتخابی) و محل اولیه برای هر کدام از آنها تعیین میگردد. سپس برای هرگره (هرالگو) یک وزنه تعیین میگردد که متناسب با مجموع حاصلضرب طول مسیرهای متصل به گره در چگالی جریان مربوطه خواهد بود. علت تعیین وزنه برای هر گره آنست که در رابطه (۳) اثر هر الگو در نماینده انتخابی، یکسان در نظر گرفته شده است حال آنکه در جایابی محل بهینه ترانس، اثر هر گره در تعیین محل ترانس با دیگر گرهها تفاوت دارد و متناسب با بار مصرفی گره مورد نظر میباشد و چون در این روش از چگالی جریان بار و پیوستگی آن استفاده شد، لذا میتوان برای هر گره، بار مناسب را شبیه سازی نمود. با توجه به موارد گفته شده رابطه تنظیم محل پستهای توزیع در هر مرحله تکرار عبارتست از:

$$\Delta P_i = \alpha \times a_i \times DIS\ (i,post) \qquad (۴)$$

که درآن میزان ΔP_i میزان حرکت ترانس بسمت گره i و a_i وزنه گره i و DIS(i,Post) فاصله ترانس

ازگره i میباشد باید توجه داشت که حاصلضرب α در a_{ij} درهیچ شرایط وبرای هیچ گرهی نباید از ۰/۱ بزرگتر باشد. رعایت این امر بمنظور همگرایی بهتر و حرکتهای منظم و بدون پرش پست‌ها در حالت تنظیم لازم است.

ابتدا ماتریس point به ابعاد ۳ × k را اختیار می‌کنیم که نشان دهندهٔ موقعیت اولیه ترانسها می‌باشد. k تعداد ترانس و سطر iام مربوط به پست i ام می‌باشد که ستون اول شماره گره ابتدا، ستون دوم شماره گره انتها و ستون سوم فاصله از گره ابتدا را نشان می‌دهد. حال بصورت تصادفی، یک گره انتخاب شده و ترانسی برای تنظیم تعیین می‌گردد که کمترین فاصله با گره فوق را داشته باشد، سپس طبق رابطه ۴ مقدار حرکت پست برنده بسوی گره مذکور بدست آمده و سطر مربوط به پست برنده در ماتریس point بر اساس میزان حرکت بدست آمده تغییر می‌کند. بعد از پایان تعداد مراحل تکرار، ماتریس point محل دقیق ترانسها را نشان می‌دهد.

۳ـ برنامه ریزی پویا (Dynamic Programming):

همانطور که از رابطه (۴) نمایان است، برای تنظیم ماتریس point نیاز به تعیین می‌نیمم فاصله گره‌های شبکه از هم و مسیر می‌نیمم می‌باشد، در این راستا از برنامه‌ریزی پویا استفاده کرده و به نتیجه مطلوب می‌رسیم. برای این منظور از یک ماتریس کمکی استفاده کرده و برای هر گره دلخواه i می‌توان فاصله و مسیر می‌نیمم، تا سایر گره‌های شبکه را بدست آوریم.

ابتدا در ستون اول ماتریس کمکی و در سه سطر اول بترتیب i ، ۰ ، i قرار می‌دهیم یعنی فاصله گره i از i صفر بوده و مسیر می‌نیمم از طریق گره i ممکن می‌شود. سپس با کمک ماتریس تلاقی، کلیه گره‌های دارای ارتباط مستقیم با گره i را مشخص کرده و در ستون دوم ماتریس کمکی قرار می‌دهیم برای هر گره در ستون دوم، سه سطر در نظر گرفته می‌شود. در سطر اول شماره گره در سطر دوم فاصله تا گره i و در سطر سوم شماره گره ماقبل در جهت رسیدن به گره i ارائه می‌گردد. سپس برای گره‌هایی که در سطرهای 2 - 3k و ستون دوم ماتریس کمکی قرار دارند عملیات تکرار و نتایج در ستون سوم درج می‌گردد لازم بذکر است که هنگام درج اطلاعات مربوط به یک گره، باید کنترل نمود که آیا گره مذکور قبلاً در ماتریس کمکی نوشته شده یا خیر و در صورت تکرار گره باید فاصله می‌نیمم تا گره iام را با هم مقایسه کرده و مسافت کمتر ملاک انتخاب می‌گردد. عملیات تا زمانی ادامه می‌یابد که هیچ گرهی در ستون مابعد ماتریس کمکی وارد نشود در ضمن باید توجه نمود که در ستون‌های ماتریس

کمکی اطلاعات مربوط به هر گره فقط یکبار درج گردد.

روشن است که عملیات زمانی پایان می‌پذیرد که شماره کلیه گره‌ها در سطرهای 2 - 3k ماتریس کمکی وارد شده باشد که به این ترتیب ضمن تعیین فاصله می‌نیمم هر گره i، جهت مسیر حرکت بسوی گره i توسط گره ماقبل درج شده و در سطر 3k، مشخص می‌گردد.

با کامل شدن ماتریس کمکی نتایج در سه سطر 1 - 3i ، 1 - 3i ، 3i ماتریس خروجی مربوط به برنامه‌ریزی پویا ذخیره می‌گردد. ماتریس خروجی این بخش از برنامه، دارای ابعاد $3n \times n$ بوده که هر سه ردیف متوالی 2 - 3k ، 1 - 3k ، 3k اطلاعات مربوط به گره k ام را نسبت به سایر گرهها نشان می‌دهد، برای روشن شدن موضوع ضمن بررسی یک شبکه نمونه، قابلیت‌های برنامه بیان می‌گردد.

۴ـ بررسی یک شبکه نمونه:

برای معرفی بهتربرنامه ، قابلیت آن راطی یک مثال بررسی می‌کنیم .

برنامه تهیه شده شامل دو بخش اصلی است که در ذیل به بررسی آنها می‌پردازیم.

۴ـ۱ ـ بخش ورودی برنامه:

ورودی برنامه بصورت گرافیکی بوده و user می‌تواند با کمک Mouse ، هر شبکه دلخواه را رسم نموده و اطلاعات مربوط به هر مسیر را، با انتخاب دو گره انتهایی آن وارد نماید. بعد از وارد نمودن اطلاعات که شامل طول و چگالی جریان هر مسیر است، مقادیر، در محلهای مناسب خود قرار می‌گیرند. برای Edit نمودن شبکه، شامل حذف و اضافه گره یا مسیر، ذخیره یا پاک نمودن اطلاعات و غیره، کلیدهایی در نظر گرفته شده که در شکلهای ۶ و ۷ نشان داده شده‌اند. در شکل ۶ یک شبکه نمونه شامل ۱۶ گره و ۲۵ مسیر همراه با اطلاعات مربوط به طول مسیر، و در شکل ۷، این شبکه همراه با چگالی جریان هر مسیر نشان داده شده است.

اطلاعات وارد شده بصورت گرافیکی، در ماتریسهای تلاقی، مسافت و چگالی جریان، ذخیره شده، بطوریکه ورودی زیر برنامه مربوط به برنامه‌ریزی پویا را تشکیل می‌دهند.

۴ـ۲ ـ بخش محاسباتی برنامه:

در این بخش از اطلاعات ورودی برنامه، استفاده شده و زیر برنامه‌های مربوط به برنامه‌ریزی

پویا و شبکه عصبی اجرا می‌گردد و در خاتمه خروجی هم در فایل ثبت شده و هم بصورت گرافیکی نشان داده می‌شود. برای شبکه نمونه فوق نتایج در جدول (۱) و شکلهای ۸الی ۱۱ نشان داده شده‌اند.

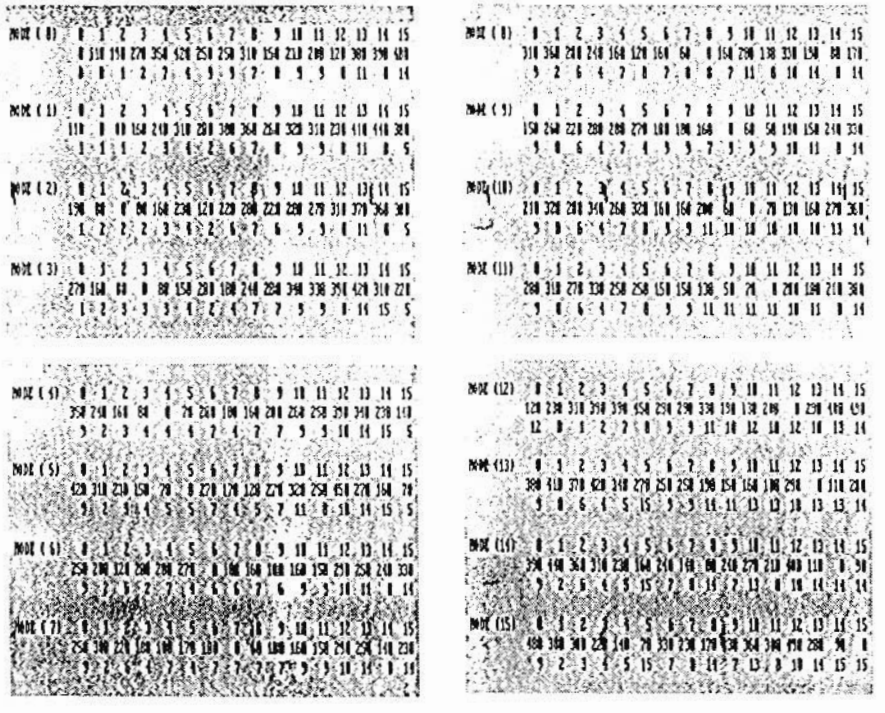
در جدول (۱)، بعنوان مثال، در سطرهای اول تا سوم، مسیر و فاصله می‌نیمم گره شماره یک از سایر گره‌ها بدست آمده است.

شکل (۸) محل اولیه ترانس را برای حالتی نشان می‌دهد که فقط یک ترانس برای شبکه مدنظر است.

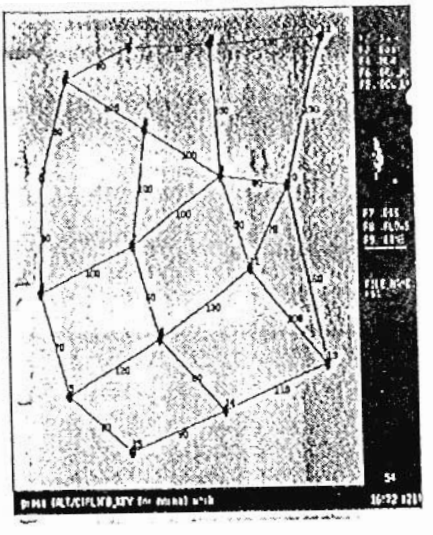
شکل (۹) محل ترانس را بعد از آموزش شبکه توسط الگوریتم **Kohonen** ، نشان می‌دهد.

شکل (۱۰) محل اولیه ترانس را برای حالتی نشان می‌دهد که دو ترانس برای شبکه مد نظر است.

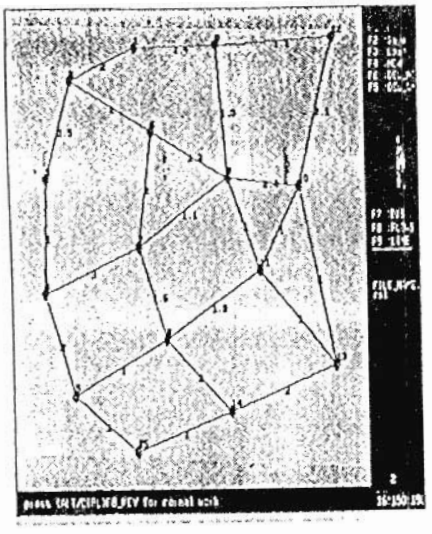
شکل (۱۱) محل نهایی ترانسها را در شبکه نشان می‌دهد.



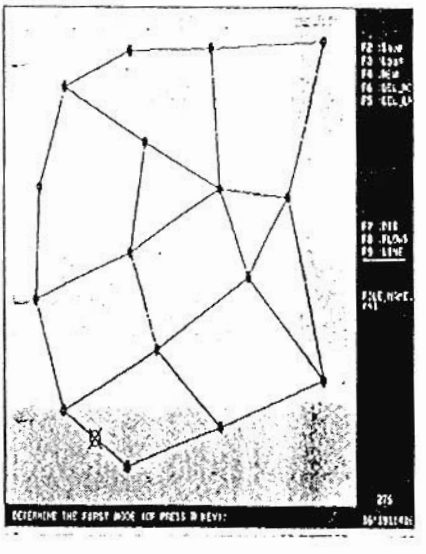
جدول (۱) خروجی برنامه ریزی پویا



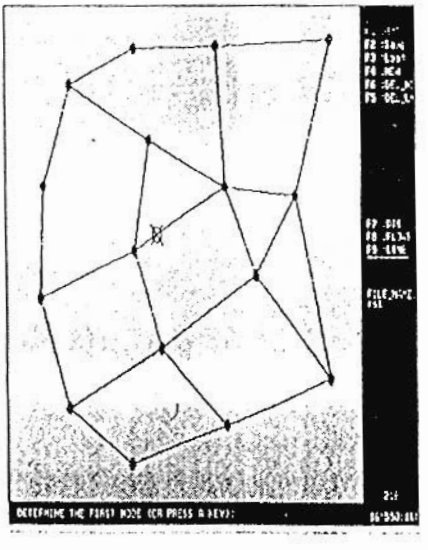
شکل (۶) ـ شبکه نمونه همراه با طول مسیر



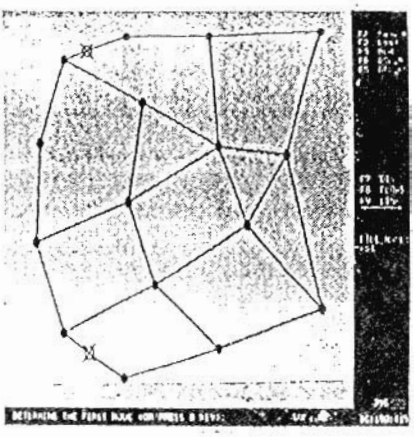
شکل (۷) ـ همراه با چگالی جریان



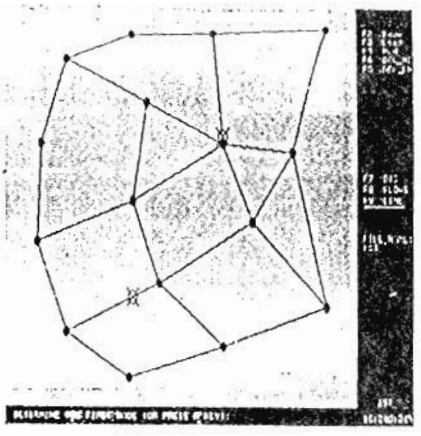
شکل (۸) ـ شبکه نمونهٔ قبل از آموزش



شکال (۹) ـ شبکه نمونه بعد از آموزش(برای یک ترانس



شکل (۱۰) ـ شبکه نمونه قبل از آموزش (برای دو ترانس)



شکل (۱۱) ـ شبکه نمونه بعد از آموزش (برای دو ترانس)

۵ـ نتیجه:

همانطور که توضیح داده شد این روش سه نقص عمده مربوط به روابـط (۱) و (۲) را رفـع می‌نماید.

عوامل مؤثر در انتخاب محل پست، فاصله و اندازه بار در محل مصرف می‌باشد، واضح است، الگوریتمی دقیق خواهد بود که عوامل فوق را بطریق مناسب در نظر بگیرد همانطور کـه در زمـینه شبکه‌های عصبی و برنامه‌ریزی پویا توضیح داده شد، این دو روش هر دو عامل فاصله و میزان بار را در انتخاب محل ترانس در نظر می‌گیرند لذا با توجه به اینکه در شبکه‌های بزرگ، امکان انتخاب محل بهینه ترانس بر اساس تجربه و همچنین استفاده از روابط (۱) و (۲) مقدور نمی‌باشد، می‌توان دقت بالای روشهای مذکور در مقاله را تأیید نمود.

۶ـ مراجع:

1 - Neural networks. Algorithm , application and programming techniques. james A.Freemen - David M.Skapura.

۲ ـ تولید و بهره‌برداری و کنترل در سیستمهای قدرت ـ ولنبرگ وود ـ ترجمه دکتر حسین سیفی

۳ ـ پایان نامه کارشناسی ارشد «تشخیص جریان تهاجمی بوسیله شبکه عصبی» ـ جمال مشتاق ـ دانشگاه صنعتی خواجه‌نصیرالدین طوسی.