بهینه سازی فضا بندی و سایز بندی مسیرهای شبکه های پاور/گراند مبتنی بر الگوریتم ساده شبکه

(sequential network simplex algorithm)

چکیده

این مقاله یک الگوریتم سریع برای بهینه سازی همزمان عرض و طول شبکه های پاور و گراند تحت محدودیتهای افت پاور و پرش گراند ارائه می کند. سایز بندی فضا که اجازه تغییر طول را می دهد باعث انعطاف بیشتر در حل مسائل عملی می شود . دو مقصود اصلی این مقاله عبارتند از ابتدا، اثبات کانوکس بودن ورژن ساده این مسئله برای توپولوژی کلی و دوم ارائه الگوریتم ساده شبکه به صورت متوالی (sequential network simplex algorithm) که می تواند این مسائل را با کارایی بالا حل کند.

نتایج عملی ضمن نشان دادن سرعت بالای این الگوریتم حاکی از کاهش حدود ۵۰ درصدی محدوده انتقال توان توسط این الگوریتم است.

۱) مقدمه

عملکرد فرکانس بالای دقیق مدارهای مجتمع با قطعات کوچک قرار گرفته در صفحات بزرگتر با لایه های متصل به هم وابسته به طراحی بهینه در تکنولوژی تحویل توان است.

چندین روش طراحی فیزیکی برای بهبود کیفیت انتقال توان موجود هست که از این میان روش سایز بندی مسیرهای انتقال توان (سیمها) نشان داده شده که یک روش موثر برای کاهش افت توان و پرش گراند بهمراه بهبود انتقال الکتریسیته می باشد. بهینه سازی توپولوژی نیز راه مفید دیگری بدین منظور است. برای حل مسائل بهینه سازی سایزبندی و توپولوژی روشهایی وجود دارد. مسئله سایزبندی سیم به صورت یک مسئله برنامه ریزی غیرخطی فرمول بندی شده و توسط روش لاگرانژین افزوده با روش نیوتن یا با روش (Steepest descent method) حل می شود.

مشکلاتی در بهینه سازی سایزبندی سیم و بهینه سازی توپولوژی شبکه های انتقال توان وجود دارد از جمله بالا بودن حجم مسئله و تعداد متغیرها و قیود و نیز غیر کانوکس بودن ذاتی مسئله که در نتیجه هیچ الگوریتم برنامه ریزی ریاضی نمی تواند بهینه سازی را تضمین کند.

برای فائق شدن به مشکلات بالا، روش feasible direction با تکرار برای مینیمم کردن محدوده سیم بندی کلی قبلا استفاده شده است. [5] روش دیگر به طور همزمان ولتاژ گره ها و جریان شاخه ها را به عنوان متغیر گرفته و مسئله را به دو بخش تقسیم می کند. بخش اول به یک مسئله

غیر خطی کانوکس تقلیل یافته و توسط سالور conjugate gradient حل می شود و بخش دوم نیز یک مسئله برنامه ریزی خطی است که به راحتی قابل حل است. خصوصیات خطی و convex بودن باعث جذاب بودن آن و سادگی حل با کارایی بالا می شود. [4]

به هر حال، مسئله برنامه نویسی غیر خطی منظم حل شده با روش conjugate gradient برای طراحی VLSI (مدارهای مجتمع بسیار بزرگ) عملا خیلی کند است. ایده مورد نظر تبدیل مسئله برنامه نویسی غیر خطی مقید به ترکیبی از مسائل برنامه نویسی خطی است که حل سریعتری خواهد داشت.

این مقاله یک الگوریتم سریع برای بهینه سازی همزمان عرضها و طول شبکه های پاور/گراند تحت محدودیتهای افت پاور و پرش گراند ارائه می دهد. سایزبندی فضا که اجازه تغییر طول را می دهد باعث انعطاف بیشتر در حل مسائل عملی می شود. دو مقصود اصلی این مقاله عبارتند از ابتدا، اثبات convex بودن ورژن ساده این مسئله برای توپولوژی کلی و دوم ارائه الگوریتم ساده شبکه به صورت تکراری (sequential network simplex algoreitm) که می تواند این مسائل را با کارایی بالا حل کند.

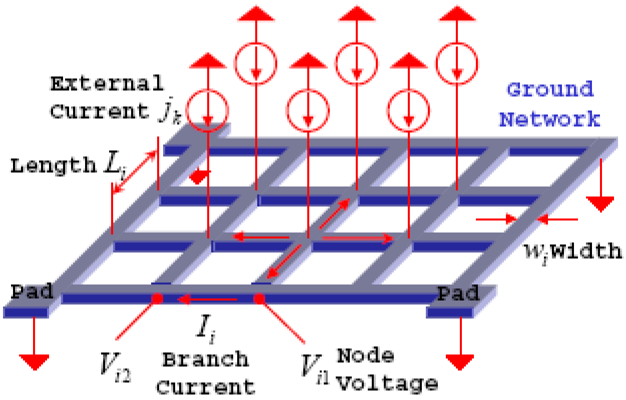
بخش دوم مقاله مروری بر مسئله و فرمولبندی آن دارد و ایده جدید مبتنی بر روش ساده شبکه network simplex algorithm در فصل سوم بحث می شود. نتایج عملی در بخش 4 و نتیجه مقاله در فصل ۵ ارائه می گردد.

۲) فرمولبندی مسئله

یک شبکه پاور/گراند G={N,B} با n گره N={1,…,n} و b شاخه B={1,...,b} مطابق شکل ۱ در نظر می گیریم. فرض شده است که میانگین جریان عبوری هر بلوک داده شده است. هر گره میانی متصل به یک منبع جریان مستقل که جریان عبوری از بلوک را مدل می کند، می باشد. سطح ولتاژ در گره i با V_i نشان داده شده و r_i, L_i, w_i به ترتیب مقاومت، طول و عرض شاخه iام را نشان می دهند. در این مقاله، فرض می کنیم که طول L_i در رنج مشخصی قابل تنظیم باشد. مقاومت هر شاخه را می توان به صورت زیر بیان کرد :

$$r_i = \rho \frac{L_i}{w_i} = \frac{V_{i1} - V_{i2}}{I_i}, \tag{1}$$

که در آن ρ مقاومت مخصوص لایه است.



شکل ۱) شبکه پاور

پس مسئله بهینه سازی فضا، مینیمم کردن ناحیه مسیرهای پاور/گراند با توجه به ولتاژها، جریانها و طول شاخه ها خواهد بود. تابع هدف می تواند به صورت زیر بیان شود:

$$F(V,I,L)=\sum_{i\in B}L_iw_i=\sum_{i\in B}\frac{\rho I_iL_i^2}{V_{i1}-V_{i2}}, \tag{2}$$

که به منظور ارضای نیازهای عملی بودن و قابلیت اطمینان شبکه پاور/گراند به محدودیتهای ذیل مقید می شود:

- محدودیت افت ولتاژ

در شبکه پاور، نوسان ولتاژ از پد پاور (power pad) تا گره های میانی (leaf node) بایستی محدود باشد که این یک حد پایین را برای سطح ولتاژ تعریف می کند.
در شبکه گراند نیز، محدودیت مشابهی حد بالای سطح ولتاژ را تعریف می کند. بنابراین محدودیتهای ولتاژ عبارتند از:

$$\begin{aligned} V_{i\in L} &\geq V_{min} \quad for\ power\ grid, \\ V_{i\in L} &\leq V_{max} \quad for\ ground\ grid, \end{aligned} \tag{3}$$

که در آن L ستی از گره های میانی و V_{max}, V_{min} حدود پایین و بالای ولتاژ هستند.

- محدودیت مینیمم عرض

مینیمم عرض یک شاخه به صورت تکنیکال با لایه فلزی که شبکه پاور/گراند را تشکیل می دهد محدود خواهد بود. این محدودیت را می توان به صورت ذیل بیان کرد:

$$w_{i \in B} = \frac{\rho L_i I_i}{V_{i1} - V_{i2}} \geq w_{min}, \quad (4)$$

که در آن $w_{\min}$ مینیمم عرض داده شده است.

- محدودیت عبور الکترونها (چگالی جریان)

طول عمر شبکه های پاور/گراند توسط معادله ذیل (Black's equation) مدل می شود:[9]

$$MTF = AJ_{avg}^{-2} exp(E_a/\kappa T) \quad (5)$$

که در آن J_{ave}, E_a, k و T به ترتیب چگالی متوسط جریان، انرژی فعالسازی، ثابت بولتزمن و دما هستند. بنابراین یک حد بالا برای چگالی جریان هر شاخه نیاز بوده و به صورت ذیل بیان می شود :

$$\begin{aligned} \frac{|I_i|}{w_i} &\leq \sigma \\ or \ |V_{i1} - V_{i2}| &\leq \rho \sigma L_i, \end{aligned} \quad (٦)$$

که در آن σ چگالی جریان برای یک ضخامت ثابت است.

- قانون جریان کرشف (KCL)

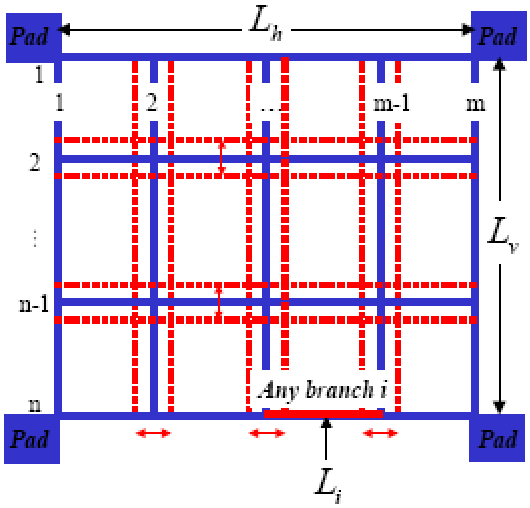
جریانهای شاخه های متصل به یک گره مقید به قانون جریان کرشف هستند:

$$\sum_{j \in N(i)} I_j = 0, \quad (7)$$

که در آن $N_{(i)}$ مجموعه شاخه های متصل به گره i هستند.

- محدودیت طول

ما به مسیرهای شبکه پاور/گراند اجازه حرکت به بالا، پایین و چپ و راست در یک رنج معین می دهیم. به هر حال فواصل مابین پدها همانطور که در شکل ۲ نشان داده شده ثابت هست.



شکل ۲) محدودیت طول مسیرها

محدودیت طول را می توان به صورت ذیل بیان کرد:

$$L_{i,lower} \leq L_i \leq L_{i,upper}, \qquad (8)$$

که در آن $L_{i,lower}, L_{i,upper}$ حدود پایین و بالای طول شاخه iام هستند. از آنجاییکه فاصله بین پدها ثابت است قیود ذیل نیز باید ارضا شوند:

$$\sum_{i=1}^{m} L_i = L_h, \quad \sum_{j=1}^{n} L_j = L_v, \qquad (9)$$

که در آن m و n تعداد مسیرهای عمودی و افقی شبکه (مسیرهای انتقال توان) می باشند. مسئله بهینه سازی فرمولبندی شده یک مسئله غیر خطی با قیود غیر خطی با متغیرهای I_i, V_i و L_i است. حل این مسئله پیچیده و از نظر محاسباتی ناکارآمد است. به این ترتیب در ادامه یک فرآیند بهینه سازی ساده با بخش بندی مسئله به دو فاز مختلف پیشنهاد می شود[4]:

۲.۱) فرآیند بهینه سازی دو فازی

در ابتدا، رسیدن به حل عملی اولیه (مقادیر اولیه) با ارضای قیود، سپس تقسیم مسئله به دو فاز P_{VL} و P_I

- **فاز اول** (P_{VL}) :

فرض کنیم که جریان شاخه ها ثابت (I_i^0) اما ولتاژ گره ها و طول شاخه ها متغیر باشد. بنابراین تابع هدف (۲) به صورت ذیل بیان می شود:

$$f_{VL}(V,L) = \sum_{i\in B} \frac{\rho I_i^{\circ} L_i^2}{V_{i1} - V_{i2}}. \tag{10}$$

که شامل محدودیتهای ذکر شده افت ولتاژ، چگالی جریان، محدودیتهای عرض مینیمم و طول مسیرها خواهد بود. باید توجه داشت که افت ولتاژ در طول یک شاخه ($V_{i1} - V_{i2}$) باید با جهت جریان I_i همعلامت باشد. قیود P_{VL} را می توان به صورت ذیل دسته بندی کرد:

$$\begin{aligned} V_i \geq V_{min} \quad or \quad V_i \leq V_{max} \\ |\, V_{i1} - V_{i2} \,| &\leq \rho\sigma L_i \\ \frac{V_{i1} - V_{i2}}{I_i^{\circ}} &\leq \frac{\rho L_i}{w_{min}} \\ L_{i,lower} \leq L_i &\leq L_{i,upper} \\ \frac{V_{i1} - V_{i2}}{I_i^{\circ}} &\geq 0. \end{aligned} \tag{11}$$

- **فاز دوم** (P_I) :

در این بخش، ولتاژ گره ها و طول شاخه ها ثابت (V_i^0, L_i^0) اما جریان شاخه ها متغیر خواهد بود. مقادیر ثابت ولتاژ گره ها و طول شاخه ها از حل P_{VL} به دست می آیند. تابع هدف را می توان به صورت ذیل بازنویسی کرد:

$$f_I(I) = \sum_{i\in B} \frac{\rho L_i^{\circ 2} I_i}{V_{i1}^{\circ} - V_{i2}^{\circ}}. \tag{12}$$

قیود مسئله شامل محدودیت ناشی از قانون جریان کرشف و محدودیت عرض مینیمم مسیرها خواهد بود. به طور مشابه افت ولتاژ در طول شاخه باید در جهت جریان عبوری از شاخه باشد.

$$\begin{aligned} \sum_{j\in B(i)} I_j &= 0 \\ \frac{I_i}{V_{i1}^{\circ} - V_{i2}^{\circ}} &\geq \frac{w_{min}}{\rho L_i^{\circ}}. \end{aligned} \tag{13}$$

۲.۲) الگوریتمی بر پایه برنامه نویسی خطی

P_{VL} یک مسئله برنامه نویسی غیر خطی با قیود خطی است. الگوریتمی بر پایه ترکیبی از برنامه های خطی توسط [1] برای جایگزینی این مسئله برنامه نویسی غیر خطی پیشنهاد شده است.
فرض کنیم که حل عملی اولیه (مقادیر اولیه) موجود باشد، بسط تیلور تابع هدف (۱۰) را حول مقادیر عملی اولیه نوشته و ترمهای ثابت و خطی آنرا نگه می داریم. تابع هدف خطی منتجه به صورت ذیل خواهد بود:

$$f(V,L) \;=\; \sum_{i\in B}\Big[\,\frac{2\rho I_i^\circ L_i^\circ}{V_{i1}^\circ - V_{i2}^\circ}L_i - \frac{\rho I_i^\circ L_i^{\circ\,2}}{(V_{i1}^\circ - V_{i2}^\circ)^2}(V_{i1}-V_{i2})\,\Big]. \tag{14}$$

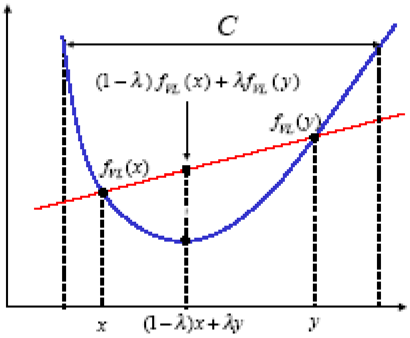
حال به جای مینیمم کردن تابع غیر خطی f_{VL} تابع خطی f را مینیمم می کنیم. از نظر تئوری، این ترکیب از برنامه های خطی همواره به یک جواب بهینه از مسئله کانوکس P_{VL} همگرا می شود.
ثابت شده است که همواره یک فاکتور محدودیت ξ وجود دارد به گونه ای که ترکیب برنامه های خطی به مینیمم مطلق مسئله همگرا شوند[1] . قیود اضافی مورد نیاز عبارتند از:

$$\begin{aligned} L_i &\geq \xi L_i^\circ \\ sign(I_i)(V_{i1}-V_{i2}) &\geq \xi sign(I_i)(V_{i1}^\circ - V_{i2}^\circ). \end{aligned} \tag{15}$$

در بخش بعدی ثابت می کنیم که تابع هدف (۱۰) یک تابع کانوکس است.

۲.۳) تابع هدف P_{VL} کانوکس است.

فرآیند بهینه سازی دو فازی نشان داده شده در بخش ۲.۱ ورژن شاده مسئله اصلی است. تابع هدف (۱۰) یعنی P_{VL} تابعی از V_i و L_i بوده و یک تابع کانوکس است. که در به صورت ذیل اثبات می شود :



شکل ۳) تابع کانوکس

فرض کنیم که C یک ست کانوکس n بعدی است ($C \in \Re^n$). تابع $f_{VL} : C \rightarrow \Re$ کانوکس است اگر و فقط اگر

$$f_{LV}((1-\lambda)x+\lambda y) \leq (1-\lambda)f_{VL}(x)+\lambda f_{VL}(y),$$
$$\forall x, y \in C, \forall \lambda \in [0,1].$$

فرض کنیم که تابع هدف $f_{VL} = \frac{L^2}{V}$ است. در نقاط x, y داریم :

$$\Rightarrow f_{VL}(x) = \frac{L_1^2}{V_1}, \qquad f_{VL}(y) = \frac{L_2^2}{V_2}.$$

$$\begin{aligned} & (1-\lambda)f_{VL}(x)+\lambda f_{VL}(y) - f_{VL}((1-\lambda)x+\lambda y) \\ = \; & (1-\lambda)\frac{L_1^2}{V_1} + \lambda\frac{L_2^2}{V_2} - \frac{[(1-\lambda)L_1+\lambda L_2]^2}{(1-\lambda)V_1+\lambda V_2} \\ = \; & \frac{[(1-\lambda)L_1^2]^2}{(1-\lambda)V_1} + \frac{[\lambda L_2^2]^2}{\lambda V_2} - \frac{[(1-\lambda)L_1+\lambda L_2]^2}{(1-\lambda)V_1+\lambda V_2} \\ = \; & \frac{\lambda(1-\lambda)[L_1V_2-L_2V_1]^2}{V_1V_2[(1-\lambda)V_1+\lambda V_2}] \geq 0 \end{aligned}$$

بنابراین تابع هدف (۱۰) با متغیرهای طول شاخه ها و ولتاژ گره ها کانوکس است.

به این ترتیب به دلیل خصوصیت کانوکس بودن P_{VL} در ادامه روش جدیدی بر پایه روش ساده شبکه ارائه می شود.

۳) ایده جدید مبتنی بر روش ساده شبکه

تحت تاثیر الگوریتم برنامه نویسی خطی به صورت متوالی روش جدیدی مبتنی بر روش ساده شبکه استفاده شده است به این دلیل که روش ساده شبکه شاید کارآمدترین الگوریتم برای حل مسائل برنامه نویسی خطی با قیود خطی است. به هر حال، اگر مسئله مینیمم هزینه انتقال ساختار شبکه ای داشته باشد حتی سریعتر می تواند توسط روش ساده شبکه حل شود.

فرآیند اصلی روش ساده شبکه به طور متوالی به صورت ذیل است:

P_{VL} و P_I را k بار تکرار کن تا زمانی که تابع هدف اصلی (۲) دیگر بهبودی نیابد. جواب هر تکرار با x^k نشان داده می شود. از آنجاییکه می دانیم که تابع هدف (۱۰) یک تابع کانوکس است می توانیم گامهای ذیل را l بار در P_{VL} تکرار کرده و به جواب مینیمم برسیم. باید توجه داشت که جواب هر زیرمسئله (P_{VL}) با x_1^k نشان داده می شود. ابتدا تابع هدف غیر خطی را از (۱۰) به (۱٤) خطی می کنیم سپس مسئله برنامه نویسی خطی را به فرمتی از یک مسئله شبکه که با الگوریتم ساده شبکه قابل حل باشد تبدیل می کنیم. برای تضمین اینکه جواب بهینه زیرمسئله ساده شبکه ((۱۱)، (۱٤)) جواب مینیمم همان مسئله اصلی ((۱۰)، (۱۱)) است ما در هر تکرار یک جستجوی خط (line search) انجام می دهیم. فرآیند را چندین مرتبه تکرار می کنیم تا دیگر بهبودی در تابع هدف (۱۰) حاصل نشود. سپس مسئله P_I را پی می گیریم.

۳.۱) تبدیل به روش ساده شبکه

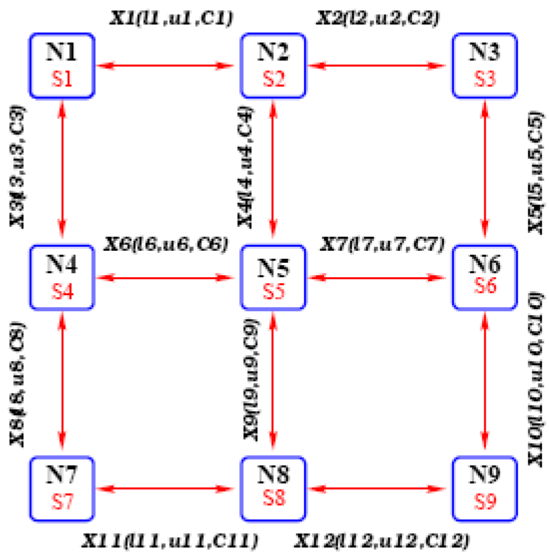
یک مسئله شبکه، مینیمم هزینه انتقال از طریق یک شبکه G={N,B} را که توسط یک ست با n گره (N) و یک ست با b شاخه (B) تعریف می شود را پیدا می کند. یک شاخه i، در ست B یک زوج مرتب (i_1, i_2) با جهت انتقال از ته i_1 به سر i_2 است. بنابراین یک مسئله شبکه را می توان به فرم ذیل نوشت

$$\begin{aligned} Minimize: &\quad \sum_{i\in B} c_i x_i \\ Subject\ to: &\quad \sum_{i\in T_n} x_i - \sum_{i\in H_n} x_i = s_n \quad \forall\ (n\in N) \\ Bounds: &\quad l_i \le x_i \le u_i \quad \forall\ (i\in B). \end{aligned}$$

- c_i هزینه پریونیتی انتقال در شاخه i است.
- x_i میزان انتقال از طریق شاخه iام است.

- T_n و H_n ستهایی از شاخه هایی هستند که به ترتیب ته ها و سرهاشان گره n می باشد.
- S_n میزان تولید/تقاضا در گره n است.
- l_i (حد پایین)، ظرفیتی است که مینیمم انتقال را از طریق شاخه i مشخص می کند.
- u_i (حد بالا)، ظرفیتی است که ماکزیمم انتقال را از طریق شاخه i مشخص می کند.

مثالی از یک مسئله شبکه در شکل ٤ نشان داده شده است.



شکل ٤) یک مسئله شبکه **نوعی**

ایده اصلی برای روش ساده شبکه [10] راه حلهای درخت پوشا (spanning tree) است. برای الگوریتم ساده شبکه، مسئله مینیمم هزینه انتقال همیشه حداقل جواب بهینه درخت پوشا دارد.
فرآیند ذیل نحوه تبدیل مسئله اصلی به مسئله شبکه را نشان می دهد. در ابتدا ما تابع هدف (١٤) را و قیود (١١) را در فرم استاندارد بازنویسی می کنیم.

$$\begin{aligned} Minimize: &\quad z = p^T x \\ Subject\ to: &\quad Ax \geq b \\ Bounds: &\quad x \geq 0. \end{aligned} \tag{16}$$

بردار x به صورت $x^T = [L^T, V^T]$ است که در آن $L = \{L_i\}$ و $V = \{V_i\}$ متغیرها را تشکیل داده و $x \in \Re^n$ است. ضریب مربوطه p^T، برابر $\frac{2\rho I_i^0 L_i^0}{V_{i1}^0 - V_{i2}^0}$ یا $\pm \frac{\rho I_i^0 L_i^{02}}{(V_{i1}^0 - V_{i2}^0)^2}$ است. هر درایه ماتریس A ضرایب V_i یا L_i در قیود (١١) هستند.

سپس متغیرهای اسلک نامنفی $x_{n+1},....,x_{n+m}$ جهت داشتن علامتهای یکسان برای قیود به سیستم اضافه می شوند. بردار جدید x^T، $[x,x_{n+1},...,x_{n+m}]^T$ است که $x \in \Re^l$ و $l = n + m$ می باشد. ماتریس جدید A، $[A,-I] \in \Re^{n \times l}$ است که در آن ماتریس I، ماتریس واحد است.
بردار p^T، $[p^T,0,...,0]$ و $p \in \Re^l$ است. بنابراین فرم کانونیک را می توان به صورت ذیل بیان کرد:

$$\begin{aligned} Minimize: \quad & z = \mathbf{p}^{\mathbf{T}}\mathbf{x} \\ Subject\ to: \quad & A\mathbf{x} = \mathbf{b} \\ Bounds: \quad & \mathbf{x} \geq 0. \end{aligned} \qquad (17)$$

باید توجه کرد که ست $\{1,2,....,l\}$ به دو ست $N = \{1,2,...,n\}$ مربوط به متغیرهای غیر اصلی و ست $B = \{n+1,n+2,...,l\}$ که مربوط به متغیرهای اصلی است تفکیک می شود. ماتریس $A_{.B} = -I$ یک ماتریس پایه نامیده می شود که خاصیت معکوس پذیری را دارد. همچنین توجه شود که برخی از قیود در (۱۱) می توانند به باندها (قیود نامساوی بزرگتر از صفر) برای ساده سازی قیود منتقل شوند. برای مثال قیود $(V_i \geq V_{\min}\ (or\ V_i \leq V_{\max})$ و $L_{i,lower} \leq L_i \leq L_{i,upper}$ می توانند به باندها منتقل شوند. مسئله برنامه نویسی خطی حالا به یک مسئله شبکه تبدیل شده و می تواند توسط روش ساده شبکه حل شود.

۳.۲) جستجوی خط (line search)

جستجوی خط نقش مهمی در بسیاری از الگوریتمهای حل مسائل برنامه نویسی غیر خطی دارد. در روش ما، مسئله غیر خطی اصلی با توالی از زیرمسئله های ساده شبکه حل می شود. به منظور تضمین اینکه جواب بهینه یک زیر مسئله ساده شبکه یک مینیمم محلی مسئله اصلی است در هر تکرار، یک جستجوی خطی نیاز است.
برای یک نقطه داده شده x_l^k، یک بردار جهت d_l و یک طول گام مناسب λ_l که منتهی به نقطه جدید $x_{l+1}^k = x_l^k + \lambda_l d_l$ می شود باید پیدا کرد. پیدا کردن طول گام λ_l شامل حل زیرمسئله برای مینیمم کردن $f_{VL}(x_l^k + \lambda d_l)$ است که یک مسئله جستجوی تک بعدی روی متغیر λ است. در مسئله ما، x_{l+1}^k، جواب بهینه جدید روش ساده شبکه با تکرار $l+1$ است.
فرآیندهای جستجوی متوالی مختلفی از قبیل جستجوی دیکوتوموس (dichotomous search)، روش بخش طلایی (golden section) و روش فیبوناچی (Fibonacci) وجود دارند که در اینجا از روش جستجوی دیکوتوموس [11] استفاده شده است. تابع هدف

$\theta(\lambda)=f_{VL}(x_l^k+\lambda d_l)$ در فاصله $[a_1,b_1]=[0,1]$ مینیمم می شود و ثابت تشخیص پذیری $2\varepsilon \succ 0$ و طول نهایی مجاز نامعینی $l_u \succ 0$ باید انتخاب شود. اجازه می دهیم که $[a_1,b_1]$ فاصله اولیه نامعینی باشد سپس به صورت ذیل به گامهای اصلی می رویم :

۱) اگر $b_k-a_k \prec l_u$ متوقف شو، نقطه مینیمم $\lambda_k=\lambda_{\min}$ در فاصله $[a_k,b_k]$ قرار دارد. در غیر این صورت λ_k و μ_k را به صورت زیر تعریف و به گام ۲ برو.

$$\lambda_k=\frac{a_k+b_k}{2}-\varepsilon \qquad \mu_k=\frac{a_k+b_k}{2}+\varepsilon$$

۲) اگر $f_{VL}(\lambda_k)\prec f_{VL}(\mu_k)$ $a_{k+1}=a_k$ و $b_{k+1}=\mu_k$ قرار بده و در غیر این صورت $a_{k+1}=\lambda_k$ و $b_{k+1}=b_k$ قرار بده $k+1$ را جایگزین k کن و برگرد به گام ۱.

بعد از بدست آوردن مقدار مینیمم $\theta(\lambda)$، نقطه x_{l+1}^k، با $x_l^k+\lambda_{\min}d_l$ برای تکرار بعدی روش ساده شبکه جایگزین می شود.

۳.۳) الگوریتم

الگوریتم ساده شبکه به صورت متوالی در جدول ۱ توصیف شده است. برخی از ملاحظات و توصیفات در ذیل بیان می شود:

۱) در ابتدا شبکه را آنالیز کرده و مقادیر اولیه L^0,V^0 و I^0 را به ترتیب برای L^k,V^k و I^k بدست آورده سپس P_{VL} و P_I را k بار تکرار می کنیم.

۲) فاز اول (P_{VL}):

مسئله را به فرمت شبکه تبدیل می کنیم. برای هر k تکرار، l تکرار برای روش ساده شبکه و جستجوی خط انجام می دهیم تا زمانی که دیگر بهبودی در f_{VL} حاصل نشود. برای هر l تکرار که از ۱ شروع می شود تابع هدف (۱٤) را با توجه به قیود (۱۱) توسط روش ساده شبکه مینیمم می کنیم. جواب بهینه (V_{l+1}^k,L_{L+1}^k)، x_{l+1}^k است. سپس جستجوی خط را انجام داده و جواب مربوطه را به عنوان (V_{l+1}^k,L_{L+1}^k) جدید برای جواب مینیمال پیدا می کنیم.

۳) فاز دوم (P_I) :

جوابهای بدست آمده از P_{VL} را جایگزین V_i و L_i کرده و تابع هدف (۱۲) را با توجه به قیود (۱۳) توسط روش برنامه نویسی ساده شبکه مینیمم می کنیم.

٤) تکرار k متوقف می شود اگر $\left|F\left(V^{k+1},L^{k+1},I^{k+1}\right)-F\left(V^k,L^k,I^k\right)\right|\le\varepsilon$

جدول ۱) الگوریتم بهینه سازی پاور/گراند مبتنی بر روش ساده شبکه **با تکرار**

Sequential Network Simplex Algorithm

Begin
$/ * \, initial \ feasible \ solution * /$
$V^k \leftarrow V^0, I^k \leftarrow I^0, L^k \leftarrow L^0;$
Repeat k
$\boxed{\mathbf{P_{VL}}:}$
Construct $\left\{ \begin{array}{l} min: \ f(V^k, L^k, I^k_{const}) \ (14) \\ s.t. \ : \ (11) \end{array} \right. ;$
$l = 1;$
$x^k_l = (V^k_l, L^k_l);$
Repeat l
Do optimization by simplex network method;
Optimal Solution:
$x^k_{l+1} = (V^k_{l+1}, L^k_{l+1});$
Do line search;
New Solution:
$x^k_{l+1} = new \ (V^k_{l+1}, L^k_{l+1});$
$l \leftarrow l + 1;$
Minimal Solution:
$x^k_l \leftarrow \ x^k_{l+1};$
until no improvement on f_{VL}
$\boxed{\mathbf{P_I}:}$
Construct $\left\{ \begin{array}{l} min: \ f_I(V^k_{const}, L^k_{const}, I^k) \ (12) \\ s.t. \ : \ (13) \end{array} \right. ;$
Do optimization by simplex network method;
$I^k \leftarrow I^{k+1};$
until $| F(V^{k+1}, L^{k+1}, I^{k+1}) - F(V^k, L^k, I^k) | \leq \varepsilon$
End

٤) نتایج عملی

الگوریتم بهینه سازی پاور/گراند مبتنی بر روش ساده شبکه پیاده سازی شده است. مجموعه ای از مدارهای شبکه ای پاور/گراند با برنامه شبیه سازی C++ مورد ارزیابی قرار گرفته شده است.

جدول ۲ شامل لیستی از نام مدارها، تعداد گره ها و شاخه های آنهاست که دو روش برنامه ریزی خطی با تکرار و روش ساده شبکه روی این مدارها از لحاظ تعداد متغیرها و قیود و مدت زمان اجرا (runtime) و درصد کاهش فضای چیپ (سطح مقطع) در مقایسه با فضای اولیه مقایسه شده اند.

از روی نتایج عملی داریم که برای مثال مدت زمان اجرای روش ساده شبکه برای مدار c60*60 که ۱۱۱٦۱ متغیر و ۲۲۲۰٤ قید دارد در مقایسه با الگوریتم برنامه نویسی خطی با تکرار، حدود ۲۵ برابر سریعتر است. با بزرگتر شدن شبکه ها مدت زمان اجرای روش ساده شبکه نسبت به برنامه

نویسی خطی با تکرار، به مراتب سریعتر می شود. به هر حال سرعت اجرای برنامه ها وابسته به توپولوژی و طراحی مسائل خواهد بود. کاهش فضا (سطح مقطع) برای همه مدارها در مقایسه با فضای اولیه حدود ۵۰ درصد است که به شدت وابسته به مقادیر اولیه (جواب اولیه) است. در روش ساده شبکه، سایز بندی طول و عرض مسیرها امکانپذیر است. طولهای کلی مابین پدها ثابت و رنج حرکتی برای هر گره ۲۰ الی ۳۰ درصد حول موقعیت اولیه است. الگوریتم ارائه شده نسبت به الگوریتم برنامه نویسی خطی با تکرار به طور مختصر فضا را بهتر کاهش می دهد. محدودیتها ناشی از توپولوژی شبکه است.

۵) نتیجه

یک الگوریتم سریع مبتنی بر روش ساده شبکه با تکرار، برای بهینه سازی همزمان عرض و طول قطعات سیم پاور/گراند شبکه ها تحت محدودیتهای قابلیت اطمینان و افت پاور و پرش گراند ارائه شد. نتایج عملی نشان داد که روش ارائه شده بسیار سریعتر از روشهای برنامه نویسی خطی با تکرار بوده و کاهش حدود ۵۰ درصدی فضای اولیه (سطح مقطع اولیه) مدارها را در بردارد.

٦) مراجع

[1] X. Tan, C. J. Richard Shi, D. Lungeanu, and L. Yuan J. Lee. Reliability-constrained area optimization of vlsi power/ground networks via sequence of linear programmings. In *36th ACM/IEEE Design Automation Conference*, pages 78–83, 1999.

[2] Salim U. Chowdhury and Melvin A. Breuer. Minimal area design of power/ground nets having graph topologies. *IEEE Trans. on Circuits and Systems*, CAS-34(12):1441–1450, December 1987.

[3] Salim U. Chowdhury and Melvin A. Breuer. Optimum design of ic power/ground nets subject to reliability constraints. *IEEE Trans. Computer-Aided Design*, 7(7):787–796, July 1988.

[4] S. Chowdhury. Optimum design of reliable ic power networks having general graph topologies. In *26th ACM/IEEE Design Automation Conference*, pages 787–790, 1989.

[5] Robi Dutta and Malgorzata Marek-Sadowska. Automatic sizing of power/ground (p/g) networks in vlsi. In *26th ACM/IEEE Design Automation Conference*, pages 783–786, 1989.

[6] Jaewon Oh and Massound Pedram. Multi-pad power/ground network design for uniform distribution of ground bounce. In *35th ACM/IEEE Design Automation Conference*, pages 287–290, 1998.

[7] H. Cai. Multi-pads single layer power net routing in vlsi circuit. In *25th ACM/IEEE Design Automation Conference*, pages 183–188, 1988.

[8] Zahir A. Syed and Abbas El Gamal. Single layer routing of power and ground networks in integrated circuits. *Journal of Digital Systems*, 6(1):1441–1450, 1982.

[9] James R. Black. Electromigration – a brief survey and some recent results. *IEEE Transactions on Electron Devices*, ED-16(4):338–347, April 1969.

[10] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows*. Prentice Hall Inc., 1993. Chapter 11.

[11] Mokhtar S. Bazaraa, Hanif D. Sherali, and C. M. Shetty. *Nonlinear Programming Theory and Algorithms*. John Wiley & Sons, Inc., 2nd edition edition, 1993. Chapter 8.