

یک کران پایین جدید برای تابع توزیع زمان تکمیل شبکه‌های پرت احتمالی

مسعود ربانی*، کامران رضائی* و محمد مهدی لطفی محمدآبادی**

گروه مهندسی صنایع، دانشکده فنی دانشگاه تهران

(دریافت مقاله: ۸۱/۱/۲۷ - دریافت نسخه نهایی: ۸۲/۱۱/۲۵)

چکیده - در این مقاله روشی جدید برای ایجاد یک کران پایین روی تابع توزیع دقیق زمان تکمیل شبکه‌های پرت احتمالی ارائه می‌شود که مبتنی بر ساده‌سازی ساختار این گونه شبکه‌ها است. مکانیزم طراحی شده با بهره‌گیری از مضاعف‌سازی برداری، ساختار شبکه را ساده می‌کند، به نحوی که تابع توزیع شبکه با استفاده انحصاری از دو عامل ضرب و پیچش قابل محاسبه باشد. نحوه انتخاب بردارهای قابل مضاعف‌سازی در روش جدید با روش مضاعف‌سازی دودین متفاوت است به گونه‌ای که باید آن را روشی کاملاً متمایز قلمداد کرد. در این روش با استفاده از مکانیزمی جدید بهترین بردارها برای مضاعف‌سازی انتخاب می‌شوند. اثبات می‌شود که تعداد مضاعف‌سازی برداری با استفاده از این روش کمینه می‌شود. تابع توزیع حاصل از این روش کران پایین تابع توزیع دقیق شبکه و البته کران بالای توزیع روش کلیندورفر و دودین است. بعد از ارائه الگوریتم این روش، کارایی آن مورد بررسی قرار می‌گیرد و با ارائه مثالهایی نتایج عددی الگوریتم با تابع توزیع دقیق شبکه و همچنین الگوریتم دودین مقایسه می‌شود.

واژگان کلیدی: شبکه، احتمالی، پرت، کران پایین

A New Lower Bound For Completion Time Distribution Function of Stochastic PERT Networks

M. Rabbani, K. Rezaie, M. M. Lotfi and M. Abadi

Assistant Professor, Industrial Engineering Department, Faculty of Engineering, Tehran University,
Former Graduate Student of Industrial Engineering

Abstract: In this paper, a new method for developing a lower bound on exact completion time distribution function of stochastic PERT networks is provided that is based on simplifying the structure of this type of network. The designed mechanism simplifies network structure by arc duplication so that network distribution function can be calculated only with convolution and multiplication. The selection of duplicable arcs in this method differs from that of Dodin's so that it must be considered a different method. In this method, best duplicable arcs are adopted using a new mechanism. It is proved that duplicating numbers is minimized by this method. The distribution function of this method is a lower bound on exact network distribution function and an upper bound on distribution function of Dodin's and Kleindorfer's methods. After the algorithm for the method is presented, its efficiency is discussed and illustration examples will be used to Compare numerical results from this method with those from exact network distribution and Dodin's method.

Keywords: Network, Stochastic, Pert, Lower Bound

**_ کارشناس ارشد

*_ استادیار

| | |
|--|---|
| Y_{ij} : متغیر تصادفی زمان فعالیت (i,j) | P : مجموعه تمام مسیرها در شبکه پرت $(-P)$ تعداد مسیرهای شبکه است |
| $F_{ij}(t)$: تابع توزیع زمان فعالیت (i,j) | $Z(\pi_i)$: طول مسیر π_i |
| $F(t)$: تابع توزیع زمان تکمیل شبکه | $G(N, A)$: گراف بدون دور، همبند و جهتدار با N گره و A بردار که دارای یک گره شروع و یک گره پایان است. |
| $D_{imp}(t)$: کران پایین روش جدید (میانگین آن μ_{imp} است) | T_i : متغیر تصادفی معرف زمان ختم گره i |
| D_{imp} : تعداد مضاعف سازیهای روش جدید | $A(i)$: $\{j \in N : (i, j) \in A\}$ ، یعنی مجموعه گره‌های مؤخر فوری i $ A(i) $ اندازه مجموعه $A(i)$ ، درجه خروجی گره i است. |
| $D(t)$: کران پایین در روش مضاعف سازی دودین (میانگین آن μ_D است) | $B(i)$: $\{k \in N : (k, i) \in A\}$ ، یعنی مجموعه گره‌های مقدم فوری i $ B(i) $ اندازه مجموعه $B(i)$ ، درجه ورودی گره i است. |
| D_d : تعداد مضاعف سازیهای روش دودین | $K(t)$: کران پایین در روش کلیندورفر (میانگین آن μ_k است) |
| D_d : تعداد مضاعف سازیهای روش دودین | $\text{Degree} : \{A(i), B(i)\}$ را درجه گره i گویند. |

۱- مقدمه

محاسبه می‌شود. زمان تکمیل شبکه معادل زمان طولانی‌ترین مسیر شبکه است، یعنی:

$$T_n = \text{Max}_{\pi \in P} \{Z(\pi)\} \quad (۳)$$

$$F(t) = \text{Pr}[T_n \leq t] = \text{Pr}[Z(\pi) \leq t; \forall \pi \in P]; t \geq 0 \quad (۴)$$

به دلایل ذیل، در اکثر موارد محاسبه طرف راست معادله (۴) دشوار است:

الف - به خاطر وجود فعالیت‌های مشترک بین دو یا چند مسیر، زمان مسیرها از نظر آماری وابستگی دارند.

ب - تعداد مسیرها با بزرگ شدن شبکه‌ها به سرعت افزایش می‌یابد. روشهای مورد استفاده برای محاسبه مشخصه‌ها را می‌توان به روشهای پرت کلاسیک، روشهای دقیق، روشهای تقریبی و روشهای شبیه‌سازی دسته‌بندی کرد. دوره کاملی از تحقیقات انجام شده تا اواخر دهه ۸۰ میلادی، در مقاله [۱] آمده است. لذا برای پرهیز از طولانی شدن مطلب تنها به مرور خلاصه‌ای از تحقیقات اخیر اکتفا می‌شود.

لو و ساماراجان [۲] روشی برای بهبود تخمین میانگین و واریانس زمان فعالیتها در پرت کلاسیک، بر مبنای تخمین هفت نقطه‌ای روش سلویج [۳] ارائه دادند. هینگ لینگ لو، هون

یک شبکه پرت، گرافی بدون دور، همبند و جهتدار با N گره و A بردار است که دارای یک گره شروع و یک گره ختم است. در یک شبکه پرت احتمالی، زمان هر بردار (فعالیت) با کمک توابع احتمال تعریف می‌شود. لذا چنانچه روشهای مناسب در برآورد مشخصه‌های این گونه شبکه‌ها در دسترس باشد، استفاده از آنها در جهت مطابقت با شرایط واقعی و حاکم بر پروژه‌ها مناسب خواهد بود.

مسئله اساسی در شبکه‌های پرت احتمالی، تعیین تابع توزیع زمان تکمیل شبکه و مشخصه‌های آن است. زمان تکمیل شبکه (T_N) متغیری تصادفی است که در اکثر موارد تعیین تابع توزیع دقیق زمان تکمیل شبکه، باید معادله (۱) محاسبه شود:

$$F(t) = \text{Pr}[T_n \leq t]; t \geq 0 \quad (۱)$$

اگر $\pi \in P$ آن‌گاه:

$$Z(\pi) = \sum (i, j) \in \pi Y_{ij} \quad (۲)$$

تابع توزیع متغیر تصادفی $Z(\pi)$ با استفاده از یک انتگرال چند گانه شامل پیچش توابع توزیع متغیرهای Y_{ij} در معادله (۲)

شیانگ و یوژانگ [۴] تعاریف a و b و نحوه تخمین m در مدل پرت کلاسیک را مورد انتقاد قرار داده و تخمینی هفت نقطه‌ای را برای میانگین و واریانس زمان فعالیتها ارائه دادند.

سروش [۶] با استفاده از مفهوم بحرانی‌ترین مسیر در یک شبکه، کرانی بالا برای تابع توزیع زمان تکمیل شبکه به دست آورد. وی همچنین در مرجع [۷] با تعریف یک تابع ریسک، مسئله مسیره‌های بحرانی را از دیدگاه نظری تصمیم‌گیری مورد توجه قرار داد. اسمیت و گروسمن [۸] تابع توزیع دقیق را برای شبکه‌هایی که توزیع زمان فعالیتها آنها ترکیبی از توابع چند جمله‌ای و دیراک بودند، تعیین کردند. روش آنها مبتنی بر استفاده از یک تبدیل خطی و همچنین ساده‌سازی ساختار شبکه با استفاده از تکنیکهای گراف بود.

دودین و سیروانسی [۹] با استفاده از توزیع مقادیر مجانبی، تقریبهایی برای توزیع زمان تکمیل شبکه و پارامترهای آن به دست آوردند. ارد [۱۰] تقریب ساده‌ای برای توزیع زمان تکمیل شبکه ارائه داد که از توزیعهای زمانی گسسته و مستقل برای فعالیتها استفاده می‌کرد. کاکس [۱۱] یک تخمین نرمال ساده برای توزیع $F(t)$ به دست آورد که بهتر از روش مسیر بحرانی عمل می‌کرد. بندل، سالومون و کارتر [۱۲] تخمینی از $F(t)$ و گشتاورهای آن، وقتی که زمان فعالیتها دارای توزیع ارلانگ بودند، ارائه کردند.

ماگوت و اسکودلارسکی [۱۳] با فرض توزیع نمایی برای زمان فعالیتها شبکه، تخمینی برای میانگین زمان تکمیل شبکه ارائه دادند. مهر و ترا، چای و پیلوتلا [۱۴] با استفاده از آماره‌های ترتیبی^۲، تخمینی را برای گشتاورهای $F(t)$ ارائه دادند. مامولو [۱۵] با استفاده از روشی موسوم به تکنیک مسیر پرت^۳ شاخصهای بحرانی فعالیتها و مسیره‌ها را تقریب زد. المغربی [۱۶] از مباحث طراحی آزمایشها^۴ برای محاسبه شاخصهای بحرانی استفاده کرد. بومن [۱۷] با استفاده از یک رویکرد شبیه‌سازی شرطی تقریبهایی را برای شاخص بحرانی فعالیتها و مسیره‌ها ارائه داد.

۲- کران پایین جدید برای محاسبه تابع توزیع زمان

تکمیل شبکه‌های پرت احتمالی

در محاسبه تابع توزیع زمان تکمیل شبکه، ساده‌ترین ساختار به شبکه‌های سری - موازی تعلق دارد. ساختار این شبکه‌ها به گونه‌ای است که فقط با استفاده از دو عملیات ضرب (برای بردارهای موازی) و پیچش (برای بردارهای سری) می‌توان شبکه را به یک بردار معادل کاهش داد.

اگر دو بردار یک شبکه به صورت سری باشند، می‌توان آنها را با یک بردار که طولش جمع این دو است، جایگزین کرد و تابع توزیع مجموع، از پیچش دو تابع توزیع مربوط به دست می‌آید. اگر دو بردار یک شبکه به صورت موازی باشند، می‌توان آنها را با یک بردار که طولش ماکزیمم این دو است، جایگزین کرد و تابع توزیع ماکزیمم، از ضرب دو تابع توزیع مربوط به دست می‌آید.

از یک دیدگاه می‌توان شبکه‌های احتمالی را در سه گروه جای داد:

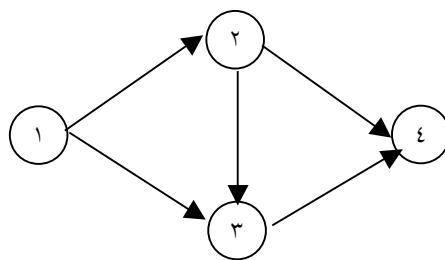
۱- شبکه کاملاً کاهش‌پذیر (سری - موازی): شبکه‌ای که با استفاده مکرر از عملیات سری و موازی به یک بردار معادل با شبکه اولیه کاهش می‌یابد. تابع توزیع بردار معادل در این نوع شبکه‌ها، معادل تابع توزیع دقیق زمان تکمیل شبکه اولیه است.

۲- شبکه کاهش‌ناپذیر: شبکه‌ای که اعمال هیچ‌یک از عملیات سری یا موازی روی آن امکان‌پذیر نباشد.

۳- شبکه کاهش‌پذیر: شبکه‌ای که با استفاده مکرر از عملیات سری و موازی به یک شبکه کاهش‌ناپذیر تبدیل شود.

۱-۲- مروری بر روش مضاعف‌سازی دودین

دودین [۱۸] الگوریتمی به نام REDUCE را برای انجام عملیات سری و موازی روی هر شبکه ارائه کرد. وی همچنین اثبات کرد یک شبکه کاملاً کاهش‌پذیر نیست اگر و فقط اگر گراف متقاطع^۵ باشد شکل (۱). روش مضاعف‌سازی، یک شبکه کاهش‌ناپذیر را به شکل کاملاً کاهش‌پذیر تبدیل می‌کند.



شکل ۱- گراف متقاطع

- $D_k(t) \geq D_{k+1}(t)$ که تابع توزیع زمان تکمیل شبکه بعد از K مضاعف‌سازی است.

دودین [۱۹] بیان می‌کند که قدرت کران پایین روش مضاعف‌سازی بستگی به تعداد مضاعف‌سازیها و شاخص بحرانیت بردارهای مضاعف شده دارد.

۲-۲- تشریح زیر بنای نظری الگوریتم محاسبه کران پایین جدید

در این بخش قضایا، تعاریف و مبانی نظری الگوریتم بهبود یافته ارائه می‌شود.

قضیه ۱: در شبکه کاهش ناپذیر $G(N,A)$ ، اگر یکی از بردارهای قابل مضاعف‌سازی ورودی به (خروجی از) گره i با $|A(i)| = n$ و $|B(i)| = n$ که $n > 1$ مضاعف شود، آن‌گاه پس از مضاعف‌سازی یکی از بردارهای عضو $\{A(i), B(i)\}$ حداکثر n عملیات موازی قابل اعمال خواهد بود.

اثبات: بدون از دست دادن کلیت فرض می‌شود بردار مورد نظر برای مضاعف‌سازی، بردار ورودی به گره‌ای مثل u باشد که $|B(u)| = n$. همچنین فرض می‌شود از بین بردارهای $\{(1,u), (2,u), \dots, (n,u)\}$ قابلیت مضاعف‌سازی داشته باشد. آن‌گاه بردار $(1,u)$ باید دارای خاصیت دو باشد شکل (۲-الف).

در نتیجه مضاعف‌سازی بردار $(1,u)$ هیچ عملیات موازی توسط بردارهای خروجی از گره u قابل تعریف نخواهد بود. طبق تعریف، دو بردار موازی دارای گره‌های شروع و ختم یکسان هستند. با توجه به اینکه بردارهای ورودی به گره u

مضاعف‌سازی یک بردار یعنی تکرار آن بردار از گره ختم یا شروع، به طوری که بردار جدید زمان و توزیع مشابه با بردار اولیه داشته باشد. اختلاف بین شبکه بعد از مضاعف‌سازی با شبکه کاهش ناپذیر اولیه، افزایش یک بردار (بردار تکراری مضاعف شده)، افزایش یک گره (گره تکراری شروع یا ختم بردار مضاعف شده) و از بین رفتن وابستگی بین بعضی از مسیرهای شامل بردار مضاعف شده است.

بردار $(i,j) \in A$ قابل مضاعف‌سازی محسوب می‌شود اگر یک یا هر دو خاصیت ذیل را داشته باشد:

۱- $(i,j) \in A$ بردار مشترک بین دو یا چند مسیر باشد و داشته باشیم: $|A(j)| > 2, |B(j)| = 1$

۲- $(i,j) \in A$ بردار مشترک بین دو یا چند مسیر باشد و داشته باشیم: $|B(i)| > 2, |A(i)| = 1$

به عنوان مثال در شکل (۱)، بردار $(۲,۱)$ دارای خاصیت یک و بردار $(۳, ۴)$ دارای خاصیت دو هستند.

روشهای مضاعف‌سازی بردارهای یک شبکه متعدد است اما الگوریتم دودین [۱۸]، بر پایه مضاعف‌سازی نخستین بردار با خاصیت یک قرار دارد.

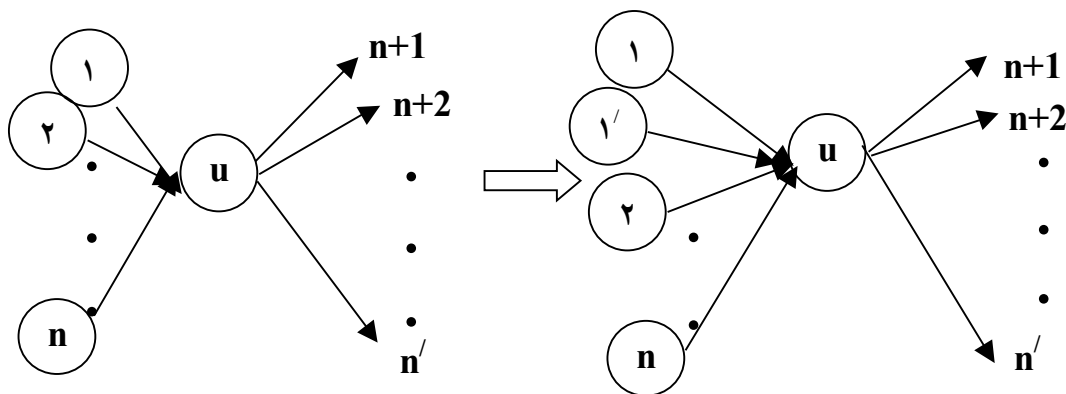
دودین [۱۹] اثبات می‌کند که:

- تابع توزیع $D(t)$ ، کران پایین تابع توزیع دقیق شبکه و کران بالای تابع توزیع روش کلیندورفر [۲۰] است، یعنی

$$K(t) \leq D(t) \leq F(t)$$

- میانگین توزیع $D(t)$ ، کران بالای میانگین دقیق شبکه و کران پایین میانگین $K(t)$ از روش کلیندورفر است یعنی

$$\mu_K \geq \mu_D \geq \mu$$



شکل (۲- الف) گره u با درجه ورودی n

عملیات چه سری و چه موازی تنها از بردار مضاعف شده قابل شروع خواهد بود. از طرفی، تنها پس از انجام یک عملیات سری ممکن است بتوان یک عملیات موازی تعریف کرد. فرض می‌شود مطابق شکل (۳) بردار مورد نظر برای مضاعف‌سازی، بردار (i, u) باشد که دارای خاصیت دو است، آن‌گاه مطابق خاصیت دو می‌توان گفت: $|A(i)| = 1$ و $|B(i)| \geq 2$. بردار (i, u) در دو حالت قابل بررسی است.

$$|B(i)| = 2 - 1$$

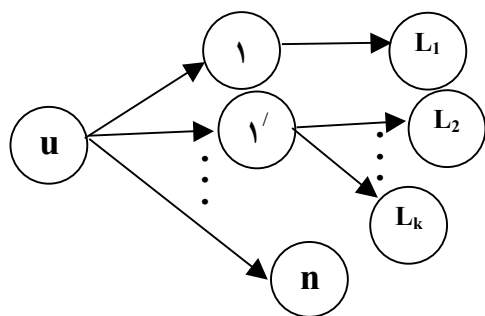
۱- الف - اگر مطابق شکل (۴ - الف)، هیچ‌یک از گره‌های مجموعه $B(i)$ ، عضو مجموعه $B(u)$ نباشند، پس از مضاعف‌سازی شکل (۴ - ب) به دست می‌آید. چون پس از انجام عملیات سری شکل (۴ - ج) دو بردار (k, u) و (l, u) حاصل می‌شوند که گره شروع آنها متعلق به $B(u)$ نیست، لذا هیچ عملیات موازی امکانپذیر نیست.

۱- ب - مطابق شکل (۵ - الف) اگر تنها یکی از گره‌های مجموعه $B(i)$ ، عضو مجموعه $B(u)$ باشد، پس از مضاعف‌سازی شکل (۵ - ب) را خواهیم داشت که بدون در نظر گرفتن نحوه قرار گرفتن سایر گره‌ها و بردارهای شبکه، دو عملیات سری $(li-iu)$ و $(ji-iu)$ و یک عملیات موازی $(ju-iu)$ تعریف می‌شود. پس از انجام عملیات مربوطه شکل (۵ - ج)

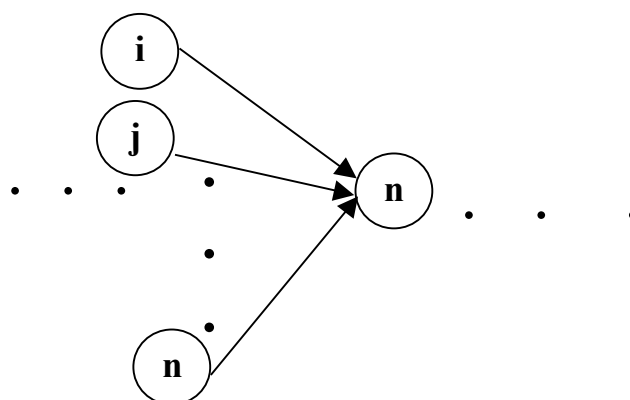
دارای گره ختم مشترک‌اند، بعد از انجام مضاعف‌سازی روی $(1, u)$ تنها ممکن است این بردارها قابلیت انجام عملیات موازی داشته باشند. در اثر مضاعف‌سازی بردار $(1, u)$ ، بردار جدید $(1, u)$ ایجاد می‌شود که با n بردار ورودی به گره u دارای گره ختم مشترک u است. لذا پس از مضاعف‌سازی $n+1$ بردار دارای گره ختم مشترک u هستند. بردارهای مزبور در شرایطی خاص حداکثر n عملیات موازی را تعریف می‌کنند و اثبات کامل می‌شود. اگر چنانچه بردار مورد نظر، بردار خروجی به گره u که $|A(u)| = n$ و $n > 1$ است باشد. از بین بردارهای $\{(u, 1), \dots, (u, n)\}$ ، اگر بردار $(u, 1)$ قابل مضاعف‌سازی باشد پس $(u, 1)$ دارای خاصیت (۱) است.

اگر مضاعف‌سازی روی $(u, 1)$ انجام شود با توجه به تعریف خاصیت (۱) در صورتی مضاعف‌سازی منجر به انجام عملیات موازی خواهد شد که بردارهای منشعب از گره 1 و گره u دارای گره ختم مشترک باشند به این ترتیب با تداوم مضاعف‌سازی بر روی شکل (۲-د) در هر نوبت حداکثر یک بار عملیات موازی مسیر است. لذا حداکثر n عملیات موازی می‌تواند انجام شود.

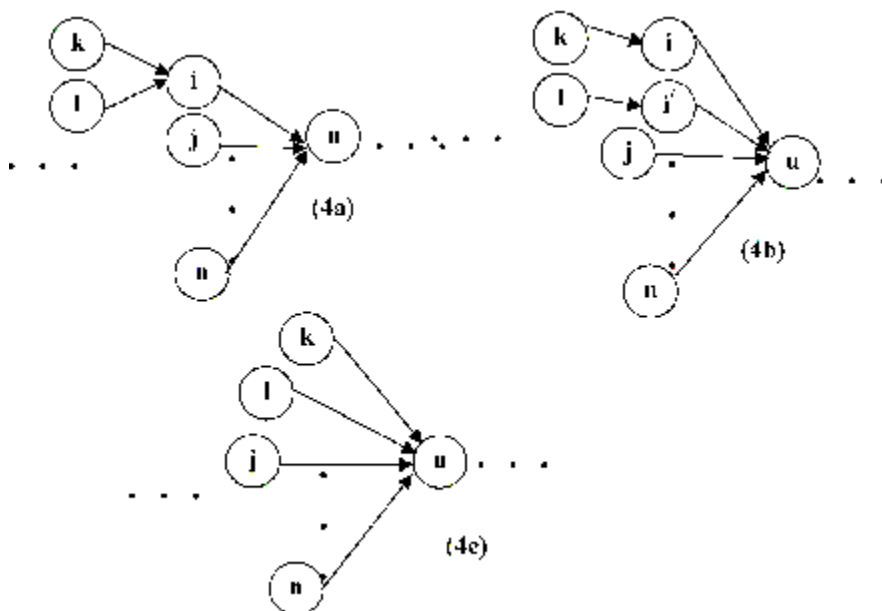
با توجه به اینکه شبکه قبل از انجام هر مضاعف‌سازی کاهش ناپذیر است، لذا بعد از انجام هر مضاعف‌سازی، هر نوع



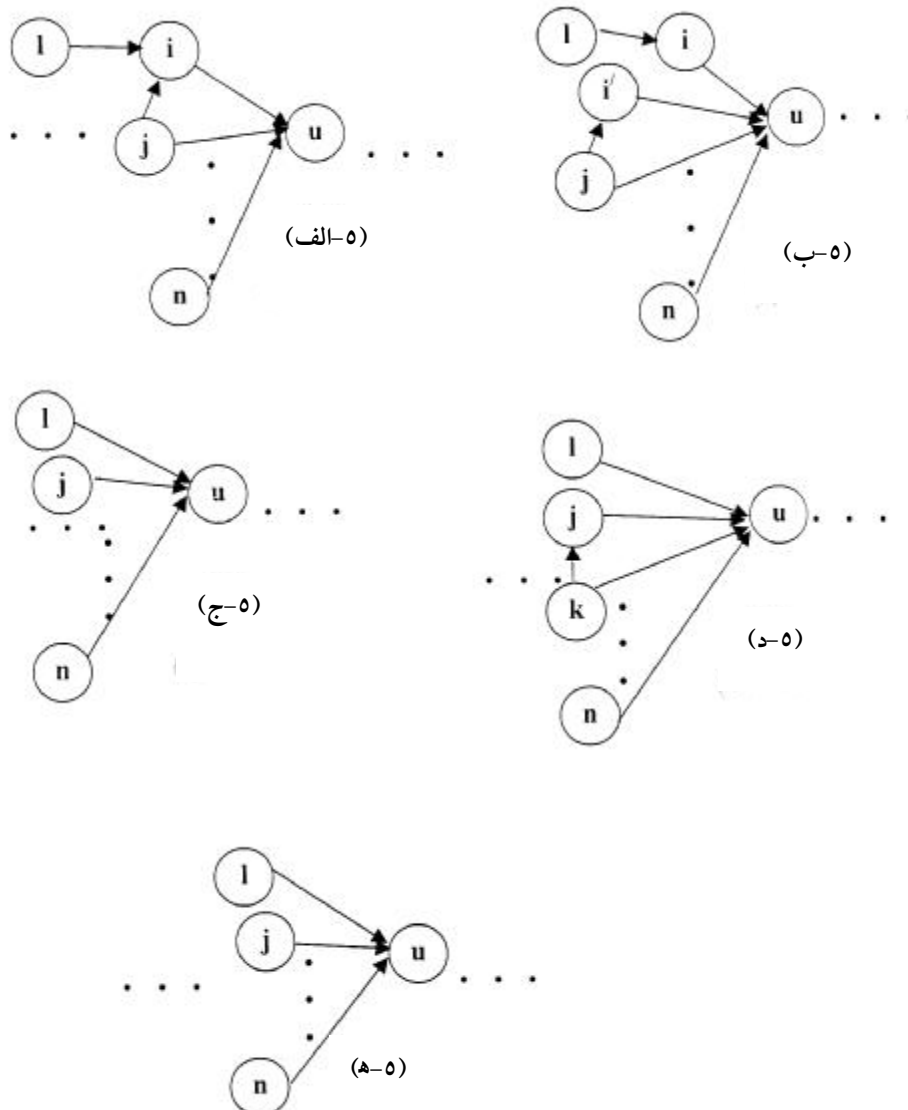
شکل (۲-ب) گره u با درجه ورودی n



شکل ۳- گره u با درجه ورودی n که بردار (i, u) قابل مضاعف‌سازی است



شکل ۴- اگر گره‌های عضو $B(i)$ متعلق به $B(u)$ نباشند



شکل ۵- اگر یکی از گره‌های عضو $B(i)$ ، متعلق به $B(u)$ باشد

بود. همچنین اگر درجه ورودی گره j بیشتر از یک باشد نیز هیچ عملیات سری قابل تعریف نخواهد بود.

اگر داشته باشیم $|A(j)|=2$ و $|B(j)|=1$ ، آن‌گاه پس از کاهش یک واحد از $|A(j)|$ خواهیم داشت:

$|A(j)|=1$ و $|B(j)|=1$ و قابلیت انجام یک عملیات سری وجود دارد.

اگر تنها گره مجموعه $B(j)$ ، گرهی مثل $k \in B(u)$ باشد شکل (۵-د)، فارغ از نحوه قرار گرفتن سایر بردارها و گره‌های شبکه، یک عملیات سری $(kj-ju)$ و یک عملیات موازی

درجه خروجی گره j یک واحد کاهش می‌یابد و گره i از شبکه حذف می‌شود.

برای اینکه عملیات موازی بعدی قابل تعریف باشد، بایستی حداقل یک عملیات سری انجام گیرد. با توجه به اینکه شبکه اولیه کاهش ناپذیر بوده است، تنها گره j می‌تواند در عملیات سری بعدی شرکت داشته باشد چون یک واحد از درجه خروجی آن کاهش یافته است. اگر درجه خروجی گره j بیشتر از دو باشد، با کاهش یک واحد از آن، عددی بزرگتر از دو خواهیم داشت و لذا هیچ عملیات سری قابل تعریف نخواهد

(ku-ku) قابل تعریف خواهد بود. این بار پس از انجام عملیات سری و موازی، درجه خروجی گره k یک واحد کاهش می‌یابد و گره z از شبکه حذف می‌شود. لذا در این حالت تنها گرهی که قابلیت تعریف عملیات سری را خواهد داشت، گره k است که مجدداً بایستی درجه ورودی و خروجی آن برای انجام عملیات سری حالت مناسب را داشته باشد. این فرایند تا آنجا ادامه می‌یابد که هیچ عملیات سری قابل تعریف نباشد. ولی اگر تنها گره مجموعه $B(i)$ ، گره I باشد، شکل (۵ - ه)، فارغ از نحوه قرار گرفتن سایر بردارها و گره‌های شبکه، یک عملیات سری $(lj-zu)$ و یک عملیات موازی $(lu-lu)$ قابل تعریف خواهد بود. سپس با حذف گره z از شبکه، امکان انجام هیچ عملیات سری یا موازی نخواهد بود و لذا متوقف می‌شویم.

۱- ج - مطابق شکل (۶)، اگر هر دو گره مجموعه $B(i)$ ، عضو مجموعه $B(u)$ نیز باشند، پس از مضاعف‌سازی (i,u) ، فارغ از نحوه قرار گرفتن سایر بردارها و گره‌های شبکه، دو عملیات سری $(ji-iu)$ و $(ki-iu)$ و دو عملیات موازی $(ju-zu)$ و $(ku-ku)$ قابل تعریف خواهد بود.

$$|B(i)| = 2 - 2$$

۲- الف - اگر هیچ یک از اعضای مجموعه $B(i)$ ، عضو مجموعه $B(u)$ نباشند، پس از مضاعف‌سازی تنها یک عملیات سری خواهیم داشت و هیچ عملیات موازی قابل تعریف نیست.

۲- ب - اگر تنها یکی از اعضای مجموعه $B(i)$ ، عضو مجموعه $B(u)$ باشد، فارغ از نحوه قرار گرفتن سایر گره‌ها و بردارهای شبکه، یک عملیات سری و یک عملیات موازی قابل تعریف خواهد بود. در این حالت نحوه مضاعف‌سازی نیز مهم است، به طوری که باید پس از مضاعف‌سازی، گره مشترک بین $B(i)$ و $B(u)$ در عملیات سری و موازی شرکت داشته باشد.

۲- ج - اگر بیش از یک عضو مجموعه $B(i)$ ، متعلق به مجموعه $B(u)$ باشند، آن‌گاه اگر گره‌های مشترک بین $B(i)$ و $B(u)$ تشکیل مجموعه M را بدهند، تمامی اعضای مجموعه M قابلیت تعریف یک یا بیشتر عملیات موازی را خواهند داشت و لذا می‌توان روند قسمت ب را برای تمام اعضای M به کار برد.

اگر بردار (i,u) دارای خاصیت (۱) باشد آن‌گاه می‌بایستی $|A(u)| > 2$ و $|B(u)| = 1$ باشد. مجدداً دو حالت $|A(u)| = 2$ و $|A(u)| > 2$ امکان رخداد دارند. در این حالت دقیقاً همانند حالت قبلی و با جایگزینی $A(u)$ به جای $B(i)$ استدلال میسر خواهد بود. باید توجه داشت که در مضاعف‌سازی کمانها، تفاوت اصلی در دو خاصیت (۱) و (۲) در گره مضاعف شده می‌باشد. اگر کمان دارای خاصیت (۱) باشد، گره انتها نیز مضاعف‌سازی می‌شود و اگر کمان دارای خاصیت (۲) گره شروع نیز مضاعف‌سازی می‌شود.

برای بررسی حالتی با حداکثر عملیات موازی پس از مضاعف‌سازی، $n=3$ در نظر گرفته می‌شود. شکل‌های (۷ - الف)، (۷ - ب) و (۷ - ج) قسمتی از یک شبکه بزرگ هستند، آن‌گاه تنها حالتی که پس از مضاعف‌سازی بردار $(3,u)$ ، حداکثر تعداد عملیات موازی (یعنی سه) را خواهیم داشت، همین سه حالت‌اند (سایر گره‌ها و بردارهای شبکه می‌توانند در هر وضعیتی قرار داشته باشند).

قضیه ۲: تعداد عملیات موازی مورد نیاز برای کاهش هر شبکه $G(N,A)$ به بردار $(1,N)$ ، مستقل از کاهش‌پذیری شبکه بوده و برابر با عدد ثابت $A-N+1$ است.

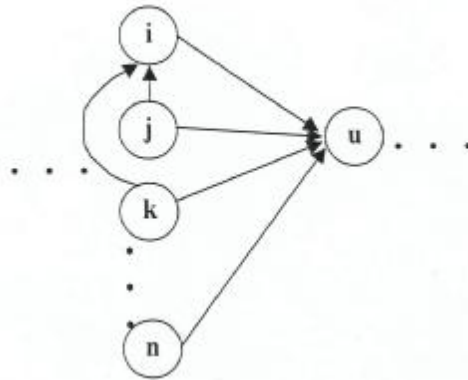
اثبات: در مقاله دودین [۱۸] آمده است.

نتیجه ۱: طبق قضیه ۲، هر زمان تعداد عملیات موازی انجام گرفته روی یک شبکه $G(N,A)$ کاهش ناپذیر معادل $A-N+1$ شود آن‌گاه:

- در شبکه حاصله هیچ عملیات موازی دیگری ممکن نیست.
- شبکه معادل کاملاً کاهش‌پذیر بوده و نیازی به مضاعف‌سازی بیشتری نیست.

قضیه ۳: اگر در یک شبکه کاهش ناپذیر تعداد عملیات موازی پس از انجام هر مضاعف‌سازی بیشینه شود، آن‌گاه تعداد مضاعف‌سازیها کمینه می‌شود.

اثبات: تعداد عملیات موازی قابل تعریف روی بردارهای شبکه پس از مضاعف‌سازی بردار (i,j) ، MO_{ij} فرض می‌شود.



شکل ۶- اگر هر دو گره عضو $B(i)$ ، متعلق به $B(u)$ باشد

طبق روش دودین باشد، آن گاه داریم

$$MOD_1 \in \{MO_{ij} | (i, j) \in A\}$$

لذا داریم:

$$MOD_1 \leq MO_1 \quad (6)$$

از طرفی داریم:

$$MO = MOD_1 + MOD_2 + \dots \quad (7)$$

پس: (1 و k مقادیری محدود، ثابت و نامشخصاند).

$MOD_1 + MOD_2 + \dots + MOD_k = MO_1 + MO_2 + \dots + MO_k$
 آن گاه طبق رابطه (6) خواهیم داشت: $1 \leq k$ و اثبات کامل می شود.

نتیجه ۳: اگر در یک شبکه کاهش ناپذیر تعداد عملیات موازی پس از انجام هر مضاعف سازی بیشینه شود، آن گاه با کاهش کامل شبکه، تابع توزیع بردار معادل $(1, N)$ ، کران بالای تابع توزیع روش کلیندوفر و دودین است.

اثبات: دودین [۱۹] اثبات می کند که:

- اگر $1 \leq k$ ، آن گاه $D_1(t) \geq D_k(t)$

- $K(t) \leq D(t)$

به وضوح با توجه به موارد فوق و همچنین نتیجه ۲، اثبات کامل می شود.

تعریف: دو بردار دارای شرایط مساوی در مضاعف سازی خواهند بود، هر گاه بر اثر مضاعف سازی هر یک، تعداد عملیات موازی حاصله یکسان باشد.

تعریف: برداری بهینه در مضاعف سازی تعریف می شود که پس از مضاعف سازی آن، حداکثر تعداد عملیات موازی نتیجه شود.

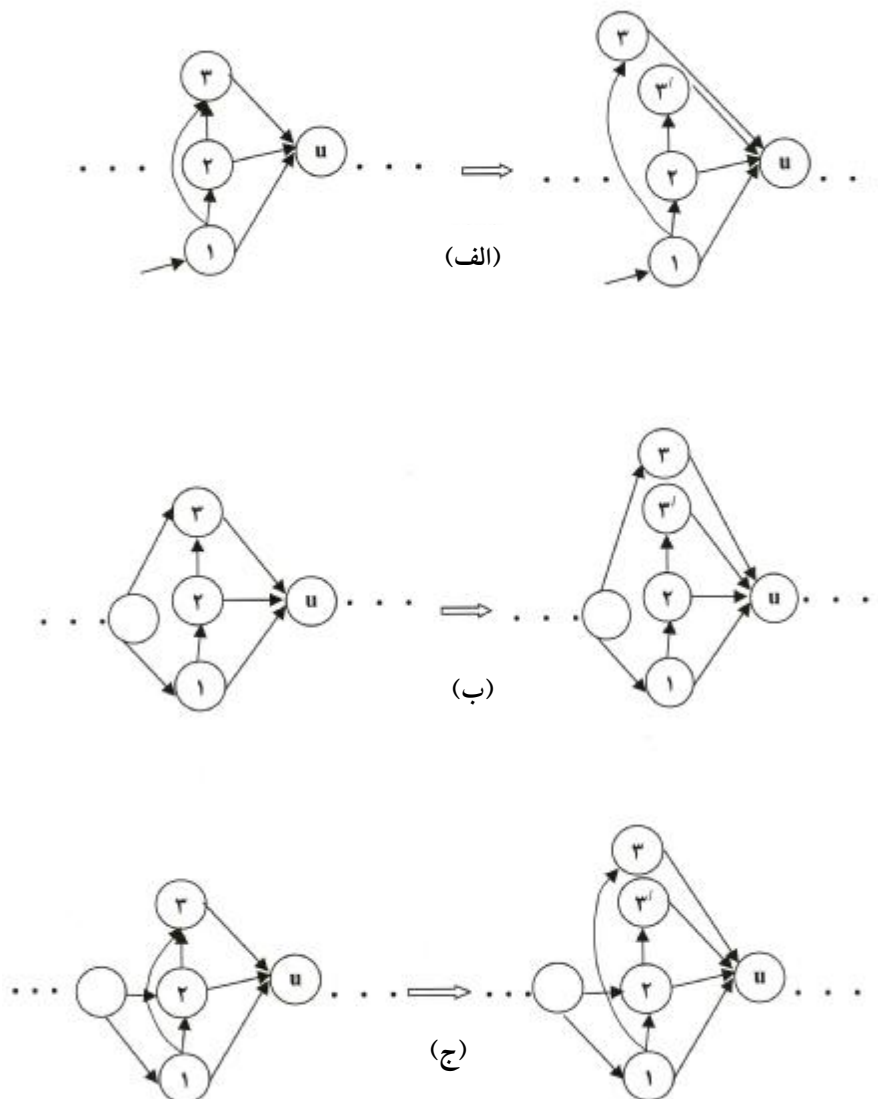
همچنین فرض می شود MO_1 تعداد عملیات موازی پس از انجام 1 امین مضاعف سازی روی شبکه و ثابت $MO = A - N + 1$ کل تعداد عملیات موازی روی شبکه باشند، آن گاه داریم:

$$MO = MO_1 + MO_2 + \dots \quad (5)$$

با توجه به قضیه ۲، طرف چپ معادله فوق برای هر شبکه $G(N, A)$ ثابت است. با توجه به ثابت و محدود بودن مقدار طرف چپ، تعداد جملات طرف راست معادله فوق نیز محدود است. هر یک از جملات طرف راست این معادله یعنی MO_1 متغیرند و $MO_1 \in \{MO_{ij} | (i, j) \in A\}$ در نتیجه با انتخاب هر یک از مقادیر MO_{ij} ، MO_1 و در نتیجه خود مقدار 1 قابل تغییر است. به وضوح اگر MO_1 ، MO_2 و ... بیشینه شوند، یعنی $MO_1 = \text{Max}\{MO_{ij} | (i, j) \in A\}$ تعداد جملات طرف راست و در نتیجه 1 کمینه می شوند و اثبات کامل می شود.

نتیجه ۲: اگر در یک شبکه کاهش ناپذیر تعداد عملیات موازی پس از انجام هر مضاعف سازی بیشینه شود، آن گاه تعداد مضاعف سازیها کران پایین تعداد مضاعف سازیهای روش دودین است.

اثبات: به وضوح طبق قضیه ۳ و اینکه تعداد مضاعف سازیها در روش دودین کمینه نیست [۱۸ و ۱۹]، نتیجه ۲ حاصل می شود. اثبات از روش دیگری نیز امکان پذیر است. طبق فرض در هر بار مضاعف سازی داریم: $MO_1 = \text{Max}\{MO_{ij} | (i, j) \in A\}$. اگر MOD_1 تعداد عملیات موازی پس از هر بار مضاعف سازی



شکل ۷- سه حالت مضاعف‌سازی بردار ورودی به گره با $|B(u)|=3$ با حداکثر عملیات موازی پس از مضاعف‌سازی (یعنی ۳)

الف - بردار مشترک بین چند مسیر باشد.

ب - بردار دارای یک یا هر دو خاصیت (۱) و (۲) باشد.

ج - بردار بهینه در مضاعف‌سازی باشد.

د - شاخص بحرانی بردار در مجموعه بردارهای بند "ج" کمترین باشد.

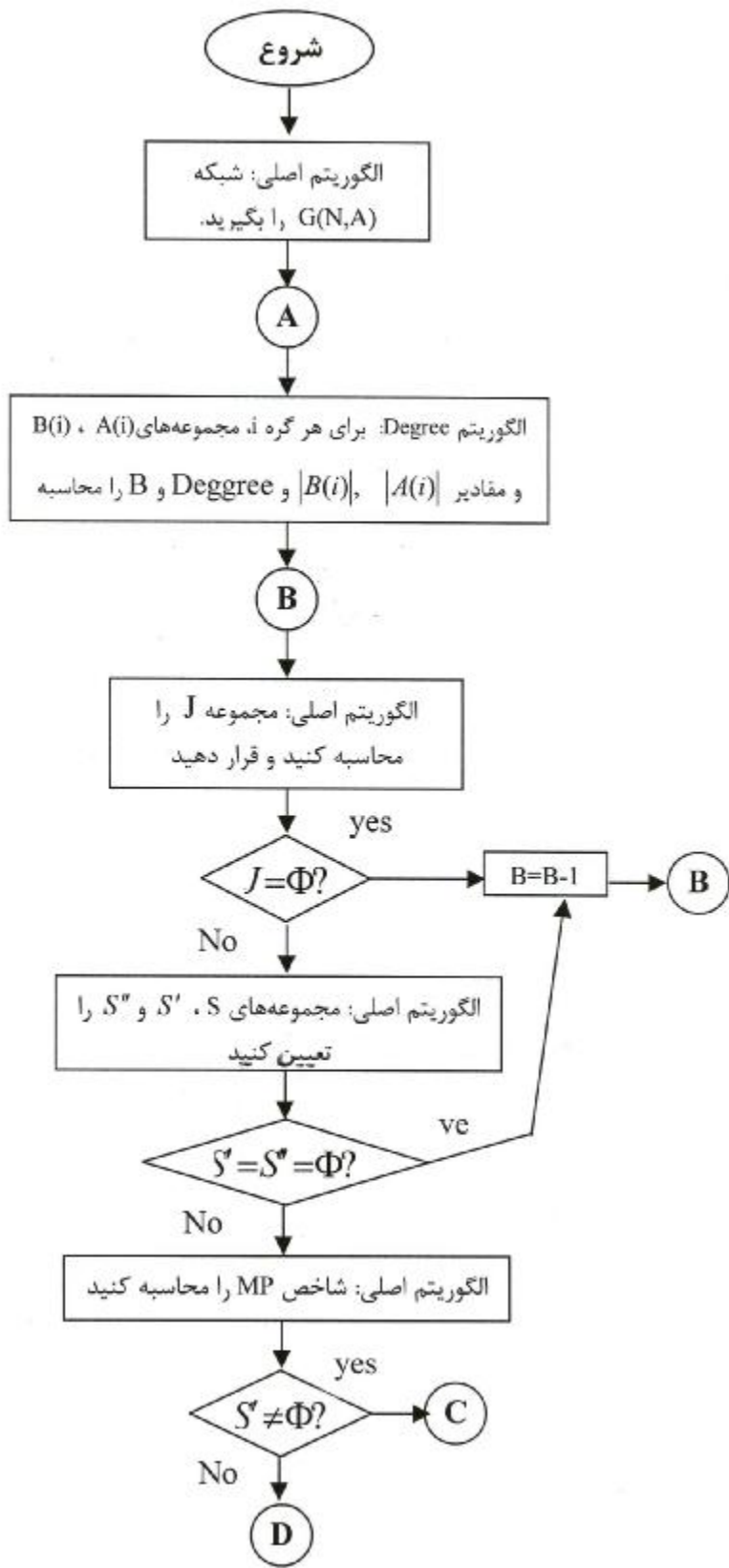
براین اساس الگوریتم پیشنهادی طراحی شده است. ساختار الگوریتم پیشنهادی در شکل (۸) ارائه شده است. الگوریتم مزبور دارای یک ساختار اصلی و چندین الگوریتم فرعی است. در ادامه ساختار الگوریتم تشریح می‌شود.

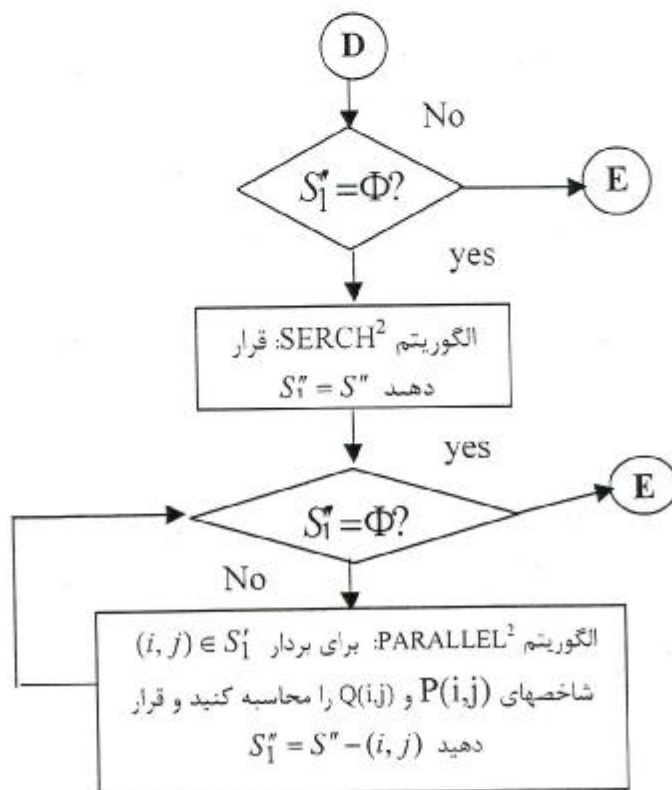
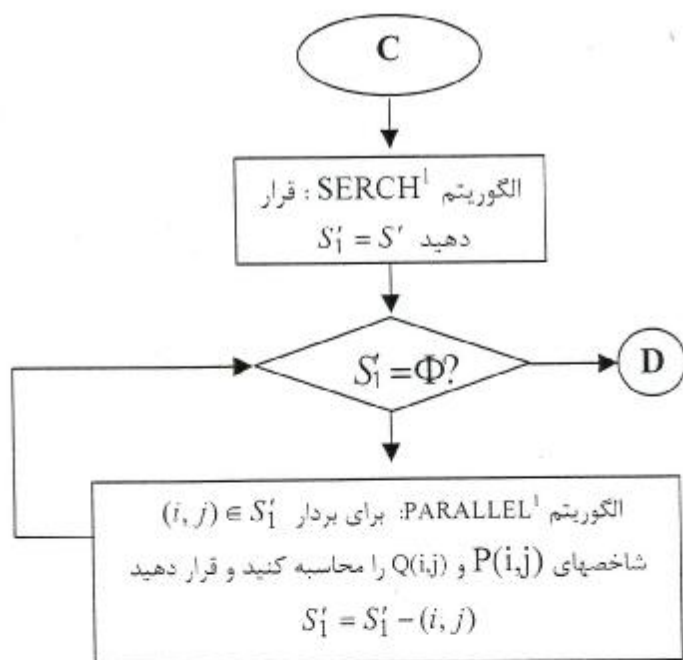
بدیهی است در سایر حالات بردار مورد نظر غیر بهینه در مضاعف‌سازی نامیده می‌شود.

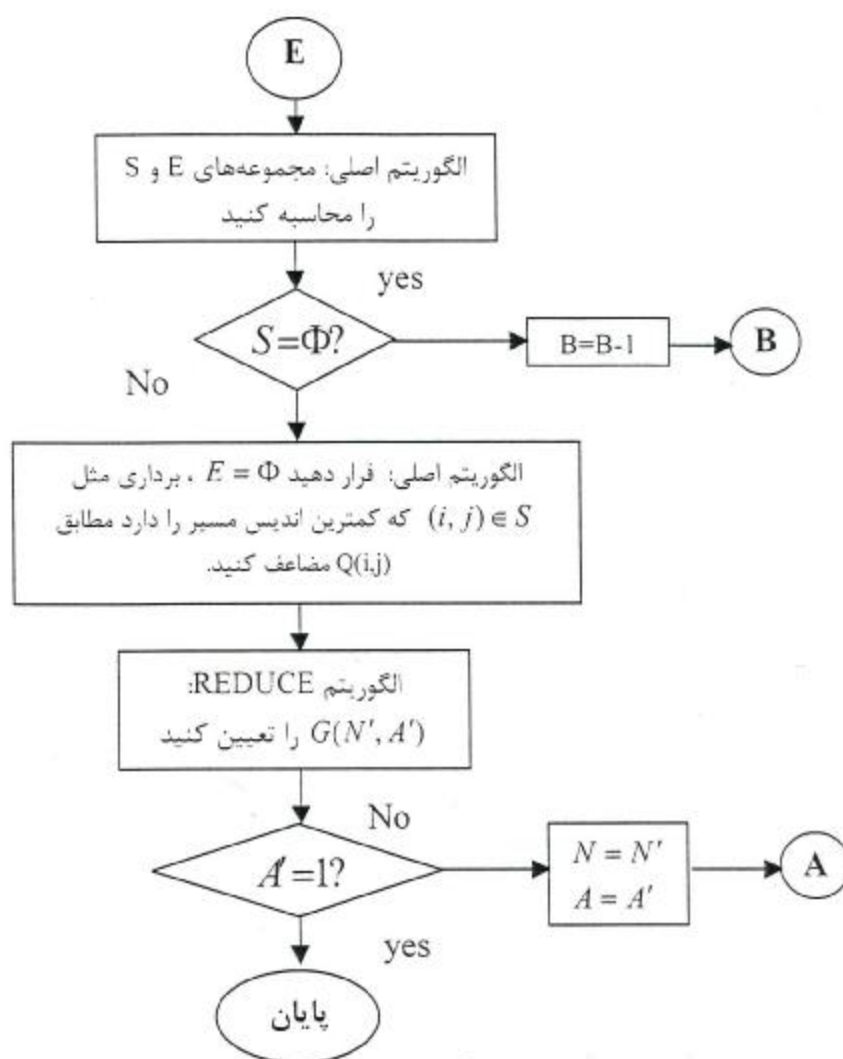
تعریف: اندیس مسیر بردار (i, j) معادل تعداد مسیرهایی است که شامل بردار (i, j) هستند.

۲-۳- الگوریتم ایجاد کران پایین جدید

با توجه به مطالب مطرح شده در مقاله و همچنین مقاله دودین [۸] به منظور مضاعف‌سازی بهینه بایستی بردار انتخابی دارای شرایط زیر باشد:







شکل ۸- نمودار جریان‌ی الگوریتم محاسبه کران پایین جدید

قابل مضاعف‌سازی باشد و $|A(j)| = 2$ (در حالتی که بردار (i,j) دارای خاصیت یک است) یا $|B(i)| = 2$ (در حالتی که بردار (i,j) دارای خاصیت دو است)، آن‌گاه یک نحوه مضاعف‌سازی برای (i,j) وجود دارد و لذا $Q(i,j) = \Phi$ باقی می‌ماند. اگر بردار (i,j) قابل مضاعف‌سازی باشد و $|A(j)| > 2$ (در حالتی که بردار (i,j) دارای خاصیت یک است) یا $|B(i)| > 2$ (در حالتی که بردار (i,j) دارای خاصیت دو است)، آن‌گاه بیش از یک نحوه مضاعف‌سازی برای بردار (i,j) وجود دارد. لذا در ادامه مجموعه $Q(i,j)$ صاحب یک عضو خواهد شد. این عضو گرهی است متعلق به $B(i)$ یا $A(j)$ که بردار

گام اول با شبکه کاهش ناپذیر $G(N,A)$ آغاز می‌شود. در صورتی که شبکه کاهش‌پذیر باشد، ابتدا با استفاده از الگوریتم REDUCE کاهش می‌یابد تا به شبکه‌ای کاهش‌ناپذیر تبدیل شود. شاخص $P(i,j)$ تعداد عملیات موازی حاصله بعد از مضاعف‌سازی بردار (i,j) است. در شروع هر بار مضاعف‌سازی این شاخص برای تمامی بردارهای شبکه مساوی صفر قرار داده می‌شود.

شاخص $Q(i,j)$ مربوط به نحوه مضاعف‌سازی بردار (i,j) است. در شروع هر بار مضاعف‌سازی این شاخص برای تمامی بردارهای شبکه مساوی تهی قرار داده می‌شود. اگر بردار (i,j)

متصل به آن در اثر مضاعف‌سازی (i,j) ، با بردار (i,j) سری خواهد شد.

پس از آن در گام ۳ با اجرای الگوریتم Degree روی شبکه، مقادیر و مجموعه‌های $A(j)$ ، $|A(i)|$ ، $B(j)$ ، $|B(i)|$ و Degree(i) مربوط به هر گره i محاسبه خواهد شد. تا زمانی که یک مضاعف‌سازی انجام نگیرد، نتایج الگوریتم Degree به قوت خود باقی می‌مانند. پس از هر بار مضاعف‌سازی الگوریتم Degree مجدداً اجرا می‌شود.

S' و S'' به ترتیب مجموعه بردارهای قابل مضاعف‌سازی دارای خاصیت یک و دو هستند. الگوریتم‌های $SEARCH^1$ و $SEARCH^2$ با این مجموعه‌ها کار می‌کنند. J مجموعه گره‌هایی از شبکه است که هنوز بررسی نشده‌اند و دارای بیشترین درجه‌اند. کلیه بردارهای متصل به گره‌های مجموعه J ، از طرفی (ورودی یا خروجی یا هر دو) که دارای بیشترین درجه است، داخل مجموعه اولیه S قرار داده می‌شوند.

MP شاخصی است برای بررسی بهینگی در مضاعف‌سازی. این شاخص همواره معادل درجه گره‌های مورد بررسی در مجموعه J است. (چون درجه گره‌های مجموعه J در هر مرحله یکسان هستند، شاخص MP در هر مرحله یک مقدار می‌گیرد). با بررسی بردارهای مجموعه S' با استفاده از زیر الگوریتم $SEARCH^1$ ، شاخص $P(i,j)$ و $Q(i,j)$ مربوط به آنها محاسبه می‌شود. با بررسی بردارهای مجموعه S'' با استفاده از زیر الگوریتم $SEARCH^2$ ، شاخص $P(i,j)$ و $Q(i,j)$ مربوط به آنها محاسبه می‌شود.

مجموعه E شامل تمامی بردارهای قابل مضاعف‌سازی است و تا زمانی که یک مضاعف‌سازی انجام نگیرد، تمامی بردارهای عضو S' و S'' داخل آن قرار داده می‌شوند. پس از هر بار مضاعف‌سازی این مجموعه خالی می‌شود.

مجموعه S سه نوع بردار را در خود جای می‌دهد: ۱- بردار یا بردارهای بهینه در مضاعف‌سازی $(P(i,j)=MP)$ ، ۲- بردار یا بردارهای غیر بهینه در مضاعف‌سازی، در صورتی که با ادامه الگوریتم برداری با بهینگی بهتر حاصل نشود. ۳- بردار یا

بردارهایی با حداکثر $P(i,j)$ در صورتی که $MP=2$ ، یعنی گره‌های مجموعه J دارای درجه مساوی دو باشند. زمانی که مجموعه S تهی نباشد، بردارهای عضو آن دارای شرایط مساوی در مضاعف‌سازی هستند، لذا برداری مثل (i,j) که کمترین اندیس مسیر را داشته باشد، انتخاب می‌شود.

سپس بردار (i,j) مطابق مجموعه $Q(i,j)$ مضاعف می‌شود و با به کارگیری الگوریتم REDUCE تمام کاهش‌های ممکن روی شبکه اعمال می‌شود. اگر شبکه حاصله از REDUCE یک بردار باشد، الگوریتم پایان می‌یابد در غیر این صورت الگوریتم به گام ۲ بر می‌گردد.

۳- بررسی مقایسه‌ای

در بررسی مقایسه‌ای سه معیار مورد توجه قرار گرفته است. الف: تعداد مضاعف‌سازی با توجه به قضایای اثبات شده در مقاله و دودین [۱۸] هرچه تعداد مضاعف‌سازیها در دستیابی به تابع توزیع زمان تکمیل شبکه کمتر باشد، تابع مزبور مقادیر نزدیکتری را به مقدار دقیق ایجاد می‌کند (مثالهای ۱ و ۲).

ب: میزان انحراف جوابهای حاصل از الگوریتم پیشنهادی در مقایسه با جواب دقیق
ج: مقایسه جوابهای حاصل از الگوریتم پیشنهادی با روش دودین

در خصوص دو مورد اخیر موارد زیر محاسبه شدند (مثال ۳).

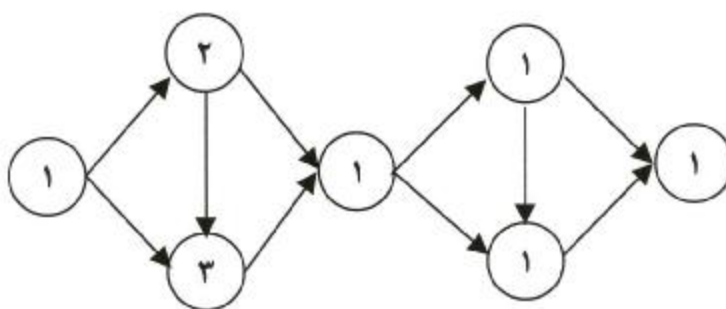
$$\left| \frac{D(t) - F(t)}{F(t)} \right| = \text{قدر مطلق انحراف جواب دودین از مقدار دقیق } (F(t))$$

قدر مطلق انحراف جواب الگوریتم پیشنهادی =

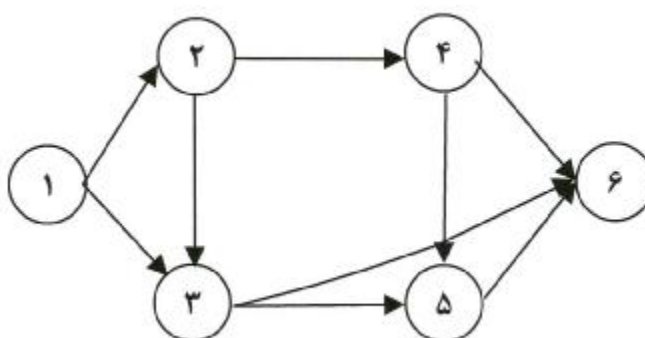
$$\left| \frac{D_{\text{Imp}}(t) - F(t)}{F(t)} \right|$$

از مقدار دقیق $(F(t))$

مثال ۱: دودین [۱۸] دو شبکه را مثال زده است شکلهای (۹) و (۱۰)، که با استفاده از الگوریتم وی، در شبکه شکل (۹)، کمانهای ۱-۲ و ۶-۷ برای مضاعف‌سازی انتخاب می‌شوند.



شکل ۹- یک شبکه کاهش ناپذیر



شکل ۱۰- یک شبکه کاهش ناپذیر

جدول ۱- نتایج عددی روش جدید در مقایسه با جواب دقیق و روش دودین برای شش شبکه

| شماره شبکه | N | A | $\frac{ D(t) - F(t) }{F(t)}$ | $\frac{ D_{Imp}(t) - F(t) }{F(t)}$ |
|------------|---|----|------------------------------|------------------------------------|
| ۱ | ۶ | ۹ | ۰/۰۹۰۰۳۵ | ۰/۰۰۷۲۳ |
| ۲ | ۶ | ۱۵ | ۰/۰۵۸۴۴۹ | ۰/۰۵۸۴۴۹ |
| ۳ | ۵ | ۹ | ۰/۰۴۲۳۸۱ | ۰/۰۰۷۰۹۴ |
| ۴ | ۷ | ۱۰ | ۰/۰۷۸۱۵۳ | ۰/۰۰۶۹۹۰ |
| ۵ | ۷ | ۹ | ۰/۰۶۶۳۳۹ | ۰/۰۴۵۲۴۳ |
| ۶ | ۷ | ۹ | ۰/۰۸۱۰۷۷ | ۰/۰۰۶۹۰۹ |

الگوریتم پیشنهادی مضاعف‌سازی بروی کمانهای ۱-۲ و ۴-۵ مضاعف‌سازی انجام می‌شود. لذا الگوریتم پیشنهادی یک مرحله مضاعف‌سازی کمتر انجام می‌دهد.
مثال ۲: شبکه‌ای دارای ۲۲ گره و ۴۰ فعالیت توسط کلیندورفر [۲۰] طراحی شده است. با استفاده از الگوریتم دودین لازمست ۱۹

طبق الگوریتم پیشنهادی نیز همین دو کمان انتخاب می‌شوند. لذا در این حالت هر دو روش یکسان خواهند بود. در شبکه شکل (۱۰) در روش دودین بر روی کمانهای ۱-۲ و ۴-۵ و ۴-۵ مضاعف‌سازی انجام می‌شود. کمان (۱-۴) در شبکه کاهش یافته پس از نوبت اول کاهش به دست می‌آید. در حالی که در

۴- نتیجه گیری

این مقاله روشی جدید را با استفاده از مضاعف‌سازی برداری، برای ایجاد کران روی تابع توزیع دقیق زمان تکمیل پروژه‌های دارای شبکه پرت احتمالی ارائه می‌کند. این روش با ارائه مکانیزمی جدید، بهترین بردارها را برای مضاعف‌سازی انتخاب می‌کند به طوری که با کمترین تعداد مضاعف‌سازی، شبکه $G(N,A)$ را به فعالیت معادل $(1,N)$ تبدیل می‌کند. اگر زمان بردارهای شبکه مستقل فرض شوند، تابع توزیع این روش، کران پایینی برای تابع توزیع دقیق زمان تکمیل شبکه و میانگین آن کران بالای میانگین دقیق زمان تکمیل شبکه است. همچنین تعداد مضاعف‌سازی در این روش کران پایین تعداد مضاعف‌سازی روش دودین است و لذا تابع توزیع آن کران بالای تابع توزیع روش دودین و کلیندورفر است.

مورد مضاعف‌سازی روی شبکه انجام گیرد تا بردار $(\mu_D, 22)$ حاصل شود در حالی که با استفاده از روش جدید نیاز به ۱۸ مورد مضاعف‌سازی خواهیم داشت که قطعاً با مقایسه نتایج عددی خواهیم داشت: $D(t) \leq D_{Imp}(t)$ و $\mu_D \geq \mu_{Imp}$ مثال ۳: در جدول (۱) نتایج مقایسه‌ای بین کارکرد روش دودین و روش جدید که هر دو قادر به ایجاد کران پایین برای تابع توزیع دقیق زمان تکمیل شبکه‌اند، ارائه شده است. به منظور مقایسه دو روش از شاخص "مجموع مربعات انحرافات از مقدار دقیق نسبت به مقدار دقیق" استفاده شده است. مطابق جدول (۱) مشاهده می‌شود که در کلیه موارد کران جدید عملکرد بهتری را در مقایسه با روش دودین ایجاد می‌کند. مکانیزم ایجاد کران جدید به گونه‌ای است که جواب حاصل برای $F(t)$ حداقل با جواب روش دودین برابر است.

واژه‌نامه

1. extreme value distribution
2. order statistics
3. PERT – path technique

4. design of experiments
5. interdictive graph

مراجع

1. Adlakha, V. G., and Kulkarni, V. G., "A Classified Bibliography of Research on Stochastic PERT Networks: 1966-1987," *INFOR*, Vol. 27, no. 3, 1989.
2. Lau, Hon-Shiang, and Somarajan, C., "A Proposal on Improved Procedures for Estimating Task-Time Distributions in PERT," *European Journal of Operational Research*, Vol. 85, pp. 39-52, 1995.
3. Selvidge, J., "Assessing the Extremes of probability Distributions by the Fractile Method," *Decision Science*, Vol. 11, pp. 493-502, 1980.
4. Lau, H. A., Lau, H., and Zhang, Y., "A Simple and Logical Alternative for Making PERT Time Estimates," *IIE Transactions*, Vol. 28, pp., 183-192, 1996.
5. Premachandra, I. M., "An Approximation of the Activity Duration Distribution in Pert," *Computers and Operations Research*, Vol. 28, pp. 443-452, 2001
6. Soroush, H. M., "The Most Critical Path in a PERT Network," *Journal of Operational Research Society*. Vol. 45, pp. 287-300, 1994.
7. Soroush, H. M., "Risk Taking in Stochastic PERT Network," *European Journal of Operation Research*, Vol. 67, pp. 221-241, 1993.
8. Schmidt, C.W. and Grossmann, I. E., "The Exact Time Distribution of a Project with Uncertain Task Durations," *European Journal of Operational Research*, Vol. 126, pp. 614-636, 2000.
9. Dodin, B., and Sirvance, M., "Stochastic Networks and the Extreme Value Distribution," *Computers and Operations Research*, Vol. 17, no. 4, pp. 397-409, 1990.
10. Ord, J. K., "A Simple Approximation to the Completion Time Distribution for a PERT Network," *Journal. of Operational Research. Society*, Vol. 42, no. 11, 1991.
11. Cox, M. A., "Simple Normal Approximation to the Completion Time Distribution for a PERT Network," *International Journal of Project Management*, Vol. 13, no. 4, pp. 265-270, 1995.
12. Bendell, A., Solomin, D., and Carter, J. M., "Evaluating Project Completion Times When Activity Times Are Erlang Distributed," *Journal of*

- the Operational Research Society*, Vol. 46, pp. 867-882, 1995.
13. Magott, J. and Skudlarski, K., "Estimating the Mean Completion Time of PERT Networks with Exponentially Distributed Durations of Activities," *European Journal of Operational Research*, Vol. 71, pp. 70-791, 1993.
 14. Mehrotra, K., Chai, J., and Pillutla, S., "A Study of Approximating the Moments of the Job Completion Time in PERT Networks," *Journal of Operations Management*, Vol. 14, no. 3, pp. 277-289, 1996.
 15. Mummolol, G., "Measuring Uncertainty and Criticality in Network Planning by PERT Path Technique," *International Journal of Project Management*, Vol. 15 no. 6, pp. 377-387, 1997.
 16. Elamghraby, S. E., "On the Criticality and Sensivity in Activity Networks," *European Journal of Operational Reserch*, Vol. 127, pp. 220-238, 2000.
 17. Bowman, R. A., "Efficient Estimation of Arc Criticalities in Stochastic Activity Networks," *Management Science*, Vol. 41 no. 1, pp. 58-67. 1995.
 18. Dodin, B., "Reducibility of Stochastic Networks," *OMEGA*, Vol. 13. no.3, pp. 223-232.
 19. Dodin, B. M. "Bounding the Project Completion Time Distribution in PERT Networks," *Operations Research*, Vol. 33. no. 4, pp. 862-881, 1985.
 20. Kleindorfer, G. B., "Bounding Distribution for a Stochastic Acyclic Network," *Operations Research*, Vol. 19. no. 7, pp. 1586-1601, 1971.

پیوست

الگوریتم Degree :

گام ۱- شبکه $G(N,A)$ را بگیرد.

گام ۲- $\forall i \in N : B(i), |B(i)|, A(i), |A(i)|$ و Degree(i) را محاسبه کنید. (مقدار Degree(i) در بخش نمادها معرفی شد.

گام ۳- قرار دهید $B = \text{Max} \{ \text{Degree}(i) \mid i \in N \}$

با توجه به اینکه الگوریتم Degree در شروع و سپس تنها بعد از هر بار مضاعف سازی اجرا می شود، شاخص B معرف حداکثر درجه در شبکه مورد بررسی است.

الگوریتم SEAECH₁

گام ۱- قرار دهید $S'_1 = S'$

گام ۲- اگر $S'_1 = \Phi$ به گام ۱۱ الگوریتم اصلی بروید، در غیر این صورت بردار $(i,j) \in S'_1$ را در نظر بگیرید.

گام ۳- الگوریتم PARALLEL₁ را اجرا کنید.

الگوریتم PARALLEL₁ :

گام ۱- اگر $|A(j)| > 2$ به گام ۲ بروید، در غیر این صورت چنین عمل کنید:

الف - اگر $A(i) \cap A(j) = \Phi$ ، قرار دهید $S'_1 = S'_1 - \{(i,j)\}$ و به گام ۲ الگوریتم SEARCH₁ بروید، و الا به گام ب بروید.

ب - اگر $|A(i) \cap A(j)| = 1$ قرار دهید $P(i,j) = P(i,j) + 1$ ، $X = A(i) \cap A(j)$

ب - ۱ - اگر هر یک از سه حالت زیر برقرار باشد، قرار دهید $S'_1 = S'_1 - \{(i,j)\}$ و به گام ۲ الگوریتم SEARCH₁ بروید.

$|A(X)| > 1 - |B(X)| > 2$ ، $A(X) \cap A(i) = \Phi$ ، $A(X) \cap A(j) = \Phi$ ، در غیر این صورت به گام ب - ۲ بروید.

ب - ۲ - اگر $|A(X) \cap A(j)| \neq \Phi$ ، قرار دهید $P(i,j) = P(i,j) + 1$ ، $S'_1 = S'_1 - \{(i,j)\}$ ، و به گام ۲ الگوریتم SEARCH₁ بروید.

در غیر این صورت به گام ب - ۳ - بروید.

ب - ۳- اگر $|A(X) \cap A(i)| \neq \Phi$ ، قرار دهید $P(i, j) = P(i, j) + 1$ و $Y = A(X) \cap A(i)$ و $X=Y$ و به گام ب - ۱ برگردید.

ج - اگر $|A(i) \cap A(j)| = 2$ ، قرار دهید $P(i, j) = P(i, j) + 2$ و چنین عمل کنید:

ج - ۱- قرار دهید $U = A(j) = \{X, Y\}$ که $X < Y$.

ج - ۲- اگر $U = \Phi$ ، قرار دهید $S'_1 = S'_1 - \{(i, j)\}$ و به گام ۲ الگوریتم $SEARCH_1$ بروید. در غیر این صورت بردار X را در نظر بگیرید و قرار دهید $U = U - \{X\}$.

ج - ۳- اگر هر یک از دو حالت زیر برقرار باشد، آنگاه اگر $U = \Phi$ ، قرار دهید $S'_1 = S'_1 - \{(i, j)\}$ و به گام ۲ الگوریتم $SEARCH_1$ بروید. در غیر این صورت قرار دهید $X=Y$ و به گام ج - ۲ بروید.

$$|A(X)| > 1 - |B(X)| > 2 -$$

در غیر این صورت به گام ج - ۴ بروید.

ج - ۴- اگر $A(X)=Y$ ، قرار دهید $P(i, j)=P(i, j)+1$ و $X=Y$ و به گام ج - ۷ بروید. در غیر این صورت به گام ج - ۵ بروید.

ج - ۵- اگر $A(X) \subset A(i)$ ، قرار دهید $P(i, j)=P(i, j)+1$ و $Z = A(X) \cap A(i)$ و به گام ج - ۶ بروید، در غیر این صورت $(A(X) \not\subset A(i))$ قرار دهید $R = \{q \in A(i), q \geq Y | A(q) = A(X)\}$. آنگاه اگر $R = \Phi$ قرار دهید $X=Y$ و به گام ج - ۲ بروید، در غیر این صورت:

- اگر $Y \in R$ و $|B(Y)| > 2$ قرار دهید $P(i, j) = P(i, j) + 1$ و $S'_1 = S'_1 - \{(i, j)\}$ و به گام ۲ الگوریتم $SEARCH_1$ بروید.

اگر $\exists q \in R : Y \in B(q)$ و $|B(Y)| = |B(q)| = 2$ و $|A(Y)| = 1$ ، قرار دهید $P(i, j) = P(i, j) + 2$ و $S'_1 = S'_1 - \{(i, j)\}$ و به گام ۲ الگوریتم $SEARCH_1$ بروید.

در غیر این دو صورت قرار دهید $X=Y$ و به گام ج - ۳ بروید.

ج - ۷- اگر $|B(X)| > 3$ یا $|A(X)| > 1$ ، قرار دهید $S'_1 = S'_1 - \{(i, j)\}$ و به گام ۲ الگوریتم $SEARCH_1$ بروید، در غیر این صورت به گام ج - ۵ بروید.

گام ۲ -

الف - اگر $|A(i) \cap A(j)| = \Phi$ ، قرار دهید $Q(i, j) = \text{Min}\{k | k \in A(j)\}$ و $S'_1 = S'_1 - \{(i, j)\}$ و به گام ۲ الگوریتم $SEARCH_1$ بروید، در غیر این صورت به گام ب بروید.

ب - اگر $|A(i) \cap A(j)| = 1$ ، قرار دهید $X = |A(i) \cap A(j)|$ ، $P(i, j) = P(i, j) + 1$ و $Q(i, j) = \{X\}$ و چنین عمل کنید:

ب - ۱- اگر هر یک از سه حالت زیر برقرار باشد، قرار دهید $S'_1 = S'_1 - \{(i, j)\}$ و به گام ۲ الگوریتم $SEARCH_1$ بروید.

$$|A(X)| > 1 - |B(X)| > 2 - A(X) \cap A(i) = \Phi -$$

ب - ۲- قرار دهید $P(i, j) = P(i, j) + 1$ و $Y = A(X) \cap A(i)$ و $X=Y$ و به گام ب - ۱ بروید.

ج - اگر $|A(i) \cap A(j)| > 1$ ، قرار دهید $U = A(i) \cap A(j) = \{u_1, u_2, \dots, u_{|U|}\}$ که $u_1 < u_2 < \dots < u_{|U|}$ و چنین عمل کنید:

ج - ۱ - قرار دهید $P_1 = P_2 = \dots = P_{|U|} = 0, n = 1$

ج - ۲ - گره $u_n \in U$ را انتخاب کرده و قرار دهید $U = U - \{u_n\}$.

ج - ۳ - اگر هر یک از سه حالت زیر رخ دهد آن گاه اگر $U = \Phi$ به گام ج - ۵ بروید، در غیر این صورت قرار دهید $n=n+1$ و به گام ج - ۲ بروید.

ج - ۴ - قرار دهید $2 - |B(u_n)| > 1 - |A(u_n)| > 1$ ، در غیر این صورت به گام ج - ۴ بروید.

ج - ۴ - قرار دهید $P_n = P_n + 1$ ، $X = u_n$ و به گام ج - ۳ بروید.

ج - ۵ - قرار دهید:

$S'_1 = S'_1 - \{(i, j)\}$ ، $Q(i, j) = \text{Min}\{u_n | P_n = \text{Max}\{P_1, P_2, \dots, P_{|U|}\}, P(i, j) = \text{Max}\{P_1, P_2, \dots, P_{|U|}\} + 1$ و به گام ۲ الگوریتم $SEARCH_1$ بروید.

الگوریتم $PARALLRL_1$ با توجه به توضیحات فصل قبل تعداد عملیات موازی را برای بردارهای مجموعه $S'_1 = S'_1$ محاسبه می‌کند. با توجه به توضیحات بخش قبل، نحوه محاسبه شاخص $P(i, j)$ و شرایط بهینگی در مضاعف‌سازی برای دو حالت $|A(j)| = 2$ و $|A(j)| > 2$ با هم متفاوت‌اند، لذا این دو حالت از هم مجزا شده‌اند. این الگوریتم تنها با نتایج الگوریتم Degree و بردارهای مجموعه S' کار می‌کند و پیچیدگی خاصی ندارد. هر زمان الگوریتم به شرایطی برسد که طبق آن یک عملیات موازی پس از مضاعف‌سازی (i, j) قابل اعمال باشد، شاخص $P(i, j)$ را یک واحد افزایش می‌دهد.

الگوریتم $SEARCH_1$:

گام ۱ - قرار دهید $S''_1 = S''_1$.

گام ۲ - اگر $S''_1 = \Phi$ به گام ۱۲ الگوریتم اصلی بروید، در غیر این صورت بردار $(i, j) \in S''_1$ را در نظر بگیرید.

گام ۳ - الگوریتم را اجرا کنید.

الگوریتم $PARALLEL_2$:

گام ۱ - اگر $|B(i)| > 2$ به گام ۲ بروید، در غیر این صورت چنین عمل کنید:

الف - اگر $B(j) \cap B(i) = \Phi$ ، قرار دهید $S''_1 = S''_1 - \{(i, j)\}$ و به گام ۲ الگوریتم $SEARCH_2$ بروید، در غیر این صورت به گام ب بروید.

ب - اگر $|B(j) \cap B(i)| = 1$ ، قرار دهید $P(i, j) = P(i, j) + 1$ و $X = B(j) \cap B(i)$

ب - ۱ - اگر هر یک از سه حالت زیر برقرار باشد، قرار دهید $S''_1 = S''_1 - \{(i, j)\}$ و به گام ۲ الگوریتم $SEARCH_2$ بروید.

ب - ۲ - اگر $|A(X)| > 2 - |B(X)| > 1$ ، در غیر این صورت به گام ب - ۲ بروید.

ب - ۲ - اگر $B(X) \cap B(i) \neq \Phi$ قرار دهید $P(i, j) = P(i, j) + 1$ و $S''_1 = S''_1 - \{(i, j)\}$ و به گام ۲ الگوریتم $SEARCH_2$ بروید. در غیر این صورت به گام ب - ۳ بروید.

ب - ۳ - اگر $B(X) \cap B(j) \neq \Phi$ ، قرار دهید $P(i, j) = P(i, j) + 1$ ، $Y = B(X) \cap B(j)$ و $X = Y$ و به گام ب - ۱ برگردید.

ج - اگر $|B(j) \cap B(i)| = 2$ ، قرار دهید $P(i, j) = P(i, j) + 2$ و چنین عمل کنید:

ج - ۱ - قرار دهید $U = B(i) = \{X, Y\}$ که $X > Y$

ج - ۲ - اگر $U = \Phi$ ، قرار دهید $S_1'' = S_1'' - \{(i, j)\}$ و به گام ۲ الگوریتم SEARCH₂ بروید. در غیر این صورت بردار X را در نظر بگیرید و قرار دهید $U = U - \{X\}$.

ج - ۳ - اگر هر یک از دو حالت زیر برقرار باشد، آنگاه اگر $U = \Phi$ ، قرار دهید $S_1'' = S_1'' - \{(i, j)\}$ و به گام ۲ الگوریتم SEARCH₂ بروید، در غیر این صورت قرار دهید $X=Y$ و به گام ج - ۲ بروید.

$$|A(X)| > 2 - , |B(X)| > 1 -$$

در غیر این صورت به گام ج - ۴ - بروید.

ج - ۴ - اگر $B(X)=Y$ ، قرار دهید $P(i,j)=P(i,j)+1$ ، $Z = B(X) \cap B(j)$ و به گام ج - ۶ بروید، در غیر این صورت $(B(X) \notin B(j))$ قرار دهید $R = \{q \in B(j), q \geq Y | B(q) = B(X)\}$ ، آنگاه اگر $R = \Phi$ قرار دهید $X=Y$ و به گام ج - ۲ بروید، در غیر این صورت:

- اگر $Y \in R$ و $|A(Y)| > 2$ قرار دهید $P(i,j)=P(i,j)+1$ ، $S_1'' = S_1'' - \{(i, j)\}$ و به گام ۲ الگوریتم SEARCH₂ بروید.

- اگر $\exists q \in R : Y \in B(q)$ و $|A(Y)| = |A(q)| = 2$ و $|B(Y)| = 1$ ، قرار دهید $P(i,j)=P(i,j)+2$ و $S_1'' = S_1'' - \{(i, j)\}$ و به گام ۲ الگوریتم SEARCH₂ بروید.

- در غیر این دو صورت قرار دهید $X=Y$ به گام ج - ۲ برگردید.

ج - ۶ - قرار دهید $X=Z$ و به گام ج - ۳ - بروید.

ج - ۷ - اگر $|A(X)| > 3$ یا $|B(X)| > 1$ ، قرار دهید $S_1'' = S_1'' - \{(i, j)\}$ و به گام ۲ الگوریتم SEARCH₂ بروید، در غیر این صورت به گام ج - ۵ - بروید.

گام ۲ -

الف - اگر $B(j) \cap B(i) = \Phi$ ، قرار دهید $Q(i, j) = \text{Min}\{k | k \in B(i)\}$ و $S_1'' = S_1'' - \{(i, j)\}$ و به گام ۲ الگوریتم SEARCH₂ بروید، در غیر این صورت به گام ب بروید.

ب - اگر $|B(j) \cap B(i)| = 1$ ، قرار دهید $X = B(j) \cap B(i)$ ، $P(i,j)=P(i,j)+1$ و $Q(i,j)=\{X\}$ و چنین عمل کنید:

ب - ۱ - اگر هر یک از سه حالت زیر برقرار باشد، قرار دهید $S_1'' = S_1'' - \{(i, j)\}$ و به گام ۲ الگوریتم SEARCH₂ بروید.

- $|A(X)| > 2 - |B(X)| > 1 -$ ، $B(X) \cap B(j) = \Phi -$ ، در غیر این صورت به گام ب - ۲ بروید.

ب - ۲ - قرار دهید $P(i,j)=P(i,j)+1$ ، $Y = B(X) \cap B(j)$ ، $X=Y$ به گام ب - ۱ بروید.

ج - اگر $|B(j) \cap B(i)| > 1$ ، قرار دهید $U = B(j) \cap A(i) = \{u_1, u_2, \dots, u_{|U|}\}$ که $u_1 > u_2 > \dots > u_{|U|}$ و چنین عمل کنید:

ج - ۱ - قرار دهید $P_1 = P_2 = \dots = P_{|U|} = 0, n = 1$

ج - ۲ - گروه $u_n \in U$ را انتخاب کرده و قرار دهید $U = U - \{u_n\}$.

ج - ۳ - اگر هر یک از سه حالت زیر رخ دهد آنگاه اگر $U = \Phi$ به گام ج - ۵ بروید، در غیر این صورت قرار دهید $n=n+1$ و به گام ج - ۲ بروید.

- $|A(u_n)| > 2 - |B(u_n)| > 1 -$ ، $B(u_n) \cap B(j) = \Phi -$ ، در غیر این صورت به گام ج - ۴ بروید.

ج - ۴ - قرار دهید $P_n = P_n + 1$ و به گام ج - ۳ بروید.
 ج - ۵ - قرار دهید:

$S_1'' = S_1'' - \{(i, j)\}$ ، $Q(i, j) = \text{Max}\{u_n | P_n = \text{Max}\{P_1, P_2, \dots, P_{|U|}\}, P(i, j) = \text{Max}\{P_1, P_2, \dots, P_{|U|}\} + 1$ و به گام ۲ الگوریتم SEARCH_2 بروید.