



پژوهش‌های نوین در تصمیم‌گیری

دوره ۵، شماره ۴، زمستان ۱۳۹۹، صص ۱۰۱-۱۱۷

نوع مقاله: پژوهشی

## تخصیص پویای سوئیچ برای کاهش هزینه در شبکه‌های مبتنی بر نرم‌افزار

احمد مرادی<sup>۱\*</sup>، علی عبدی سیدکلایی<sup>۲</sup>، سید امین حسینی سنو<sup>۳</sup>

۱- استادیار، گروه علوم کامپیوتر، دانشکده علوم ریاضی، دانشگاه مازندران، بابلسر، ایران

۲- دانشجوی دکتری، مهندسی کامپیوتر، دانشگاه فردوسی مشهد، مشهد، ایران

۳- دانشیار، گروه مهندسی کامپیوتر، دانشکده مهندسی، دانشگاه فردوسی مشهد، مشهد، ایران

تاریخ پذیرش: ۱۳۹۹/۰۵/۱۱

تاریخ ارسال: ۱۳۹۸/۱۱/۳۰

### چکیده

شبکه‌های مبتنی بر نرم‌افزار، با جداسازی سطح کنترل از سطح داده در تجهیزات شبکه همانند سوئیچ و مسیریاب مزایای بسیار زیادی همچون قابلیت برنامه‌ریزی بیشتر سطح کنترل، استقلال بیشتر شرکت‌های تولیدکننده تجهیزات شبکه، امکان مجازی‌سازی شبکه، کاهش هزینه‌های عملیاتی و... را به ارمغان آورده‌اند. به‌کارگیری یک کنترلر در شبکه معایبی به همراه دارد. ازجمله اینکه، ترافیک شبکه در بیشتر بخش‌ها شناخته‌شده بیش‌ازحد خواهد شد و کنترلر ممکن است در حالت سرریز قرار گیرد. بنابراین استفاده از چندکنترلر موجب تعریف مسئله‌ای به نام مسئله مکان‌یابی کنترلرها خواهد شد که در آن تعداد و مکان نصب کنترلرها، اهمیت می‌یابد. در این مسئله، هدف یافتن مکان و تعداد بهینه کنترلرهای موردنیاز در شبکه است که قیدهای مشخصی را برآورد و هزینه پیاده‌سازی شبکه را حداقل سازد. در این مقاله، به‌منظور حل مسئله مکان‌یابی کنترلر الگوریتمی بر اساس تخصیص پویای سوئیچ پیشنهاد و عملکرد آن در مقایسه با روش‌های حل موجود و بر روی دسته وسیع و متنوعی از نمونه‌ها مورد ارزیابی قرار گرفته است. نتایج حاصل از محاسبات بر روی نمونه‌ها نشان از برتری الگوریتم تخصیص پویای سوئیچ در زمان اجرا برای تمام نمونه‌ها و دریافتن حداقل هزینه پیاده‌سازی در برخی از نمونه‌ها با اندازه‌های کوچک و متوسط حکایت دارد. همچنین در شبکه‌های با اندازه بزرگ، الگوریتم تخصیص پویای سوئیچ چه در زمان اجرا و چه در هزینه پیاده‌سازی از برتری قطعی برخوردار است.

**کلیدواژه‌ها:** شبکه تعریف‌شده با نرم‌افزار، مکان‌یابی کنترلر، تخصیص پویا



## ۱- مقدمه

شبکه‌های سنتی به دلیل عدم انعطاف‌پذیری کافی مقرون‌به‌صرفه نیستند و برای پاسخگویی به نیازهای اینترنت فعلی مناسب نخواهند بود [۱]. بنابراین ظهور نسل جدیدی از شبکه که بتواند پاسخگو نیازها و دارای قابلیت انعطاف‌پذیری بالا باشد، مطرح می‌شود. شبکه مبتنی بر نرم‌افزار<sup>۱</sup> امکان دست یافتن به یک شبکه قابل برنامه‌ریزی با جدا کردن سطح کنترل از سطح داده برای بهبود کارایی شبکه را فراهم می‌کند [۲]. این جداسازی مزایایی همچون مدیریت ساده شبکه، بهبود کارایی شبکه و ایجاد نوآوری در شبکه را فراهم می‌کند. سطح کنترل، اطلاعات لازم برای مسیریابی در شبکه را فراهم می‌کند. سطح داده نیز وظیفه انتقال بسته‌ها از درگاه ورودی به درگاه خروجی بر اساس اطلاعات درج‌شده در جدول مسیریابی خود را بر عهده دارد. در شبکه مبتنی بر نرم‌افزار سطح جداشده کنترل در سرور یا برنامه‌ای به نام کنترلر قرار می‌گیرد. سطح داده نیز در سوئیچ یا مسیریاب به‌عنوان ارسال‌کننده باقی می‌ماند. سطح کنترل مدیریت سطح داده را بر عهده دارد. پیدایش شبکه مبتنی بر نرم‌افزار توجه تعداد زیادی از دانشگاه‌ها و صنایع را به پیاده‌سازی آن در زیرساخت‌های ارتباطی خود معطوف ساخته است [۳]. روی این شبکه‌ها، روش‌های پیکربندی، اغلب ساده‌تر و دقیق‌ترند و امکان بهره‌گیری بیشتر از زیرساخت‌های فیزیکی فراهم آمده است [۴]. گروهی از اپراتورهای شبکه، ارائه‌دهندگان خدمات شبکه و تولیدکنندگان تجهیزات شبکه، سازمانی به نام ONF<sup>۲</sup> [۵] ایجاد کردند تا به ترویج شبکه‌های مبتنی بر نرم‌افزار و ارائه یک پروتکل ارتباطی استاندارد برای این شبکه بپردازند.

در شبکه‌های مبتنی بر نرم‌افزار، کنترلر اغلب مسئول انتشار هر جریان در شبکه است که این عمل را با تخصیص جریان‌های ورودی به سوئیچ‌ها انجام می‌دهد [۶]. این مهم، نقشی محوری به کنترلر بخشیده است؛ چراکه به‌واسطه آن می‌توان دانش کاملی از شبکه را در بهینه‌سازی مدیریت جریان و پشتیبانی از نیاز کاربر به خدمت گرفت [۷].

به‌کارگیری یک کنترلر در شبکه مبتنی بر نرم‌افزار معایبی به همراه دارد. از جمله اینکه، ترافیک شبکه در بیشتر بخش‌های شناخته‌شده بیش‌ازحد خواهد شد و کنترلر ممکن است در حالت سرریز قرار گیرد. بنابراین استفاده از چند کنترلر موجب تعریف مسئله‌ای به نام مسئله مکان‌یابی کنترلرها خواهد شد که در آن تعداد و مکان نصب کنترلرها، اهمیت می‌یابد. در این



مسئله، هدف تعیین مکان کنترلرها و تعداد موردنیاز آنها در شبکه است. به طوری که بتوان حداقل هزینه را برای مجموع هزینه‌های نصب کنترلر در مکان‌ها، هزینه‌های اتصال سوئیچ‌ها به کنترلرها و نیز هزینه‌های اتصال کنترلرها به یکدیگر صرف کرد [۸]. مسئله مکان‌یابی کنترلر مشابه مسئله مکان‌یابی تسهیلات [۹] [۱۰] است و از نظر پیچیدگی محاسباتی، این مسئله به کلاس مسائل NP-سخت تعلق دارد [۱۱]. در این مقاله، به منظور حل مسئله از الگوریتمی موسوم به تخصیص پویا سوئیچ استفاده شده است. این الگوریتم از روش خاموش کردن کنترلرهای فعال برای رسیدن به هدف استفاده می‌کند تا به توپولوژی کارآمدی از شبکه در یک‌زمان قابل قبول دست یابد.

ادامه این مقاله به شرح ذیل سازمان‌دهی شده است: بخش ۲، پیشینه موجود از روش‌های حل مسئله را مرور می‌کند. بخش ۳ به تشریح یک مدل برنامه‌ریزی صحیح استاندارد از مسئله پرداخته است. بخش ۴، به معرفی الگوریتم پیشنهادی مبتنی بر تخصیص پویا سوئیچ و بحث در ساختار آن اختصاص یافته است. بخش ۵ به تحلیل محاسباتی الگوریتم و بررسی عملکرد آن در مقایسه با یک حل‌کننده استاندارد مدل‌های برنامه‌ریزی صحیح، پرداخته است. در نهایت نتیجه‌گیری حاصل از محاسبات به دست آمده در بخش ۶ بیان شده است.

## ۲- تحقیقات مرتبط

در این بخش، پژوهش‌های انجام شده روی مسئله مکان‌یابی کنترلر مرور می‌شود. مقاله [۱۲] از الگوریتم ابتکاری<sup>۲</sup> برای حل مسئله مکان‌یابی کنترلر در شبکه‌های با مقیاس بزرگ بهره برد. در الگوریتم پیشنهادی محدودیت‌های مربوط به ظرفیت کنترلرها و بار سوئیچ‌ها در نظر گرفته شد. در [۱۳] روشی برای مکان‌یابی کنترلر در شبکه گسترده مبتنی بر نرم‌افزار ارائه شد. در این مقاله، کاهش تأخیر زمانی شامل تأخیر زمانی نهایی و تأخیر زمانی صف کنترلرها مورد بررسی قرار گرفت. برای کاهش تأخیر زمانی نهایی از مفهوم پارتیشن‌بندی شبکه استفاده شد. در الگوریتم پیشنهادی، از تقسیم‌بندی بر اساس خوشه استفاده شد تا در هر قسمت تضمین شود که کمترین تأخیر نهایی بین کنترلرها و سوئیچ‌ها به دست آید. برای کاهش تأخیر صف کنترلرها نیز با به‌کارگیری تعداد مناسب کنترلر در هر زیر شبکه سعی در کاهش این تأخیر گردید. برخلاف دو پژوهش قبل، در مقاله [۱۴] قابلیت اطمینان در شبکه‌های مبتنی بر



نرم افزار مورد توجه قرار گرفت. در این تحقیق علاوه بر قابلیت اطمینان و در نظر گرفتن ظرفیت کنترلرها، راه کاری برای پیش بینی خرابی کنترلرها ارائه گردید تا از افزایش شدید تأخیرها و قطعی ارتباط جلوگیری شود. برای رسیدن به این اهداف از برنامه ریزی خطی عدد صحیح مختلط<sup>۴</sup> [۱۵] استفاده شد. تابع هدف حداقل کردن حداکثر مجموع تأخیر از سوئیچ به نزدیک ترین کنترلر دارای ظرفیت کافی (اولین کنترلر مرجع) و تأخیر از اولین کنترلر مرجع به نزدیک ترین کنترلر دارای ظرفیت کافی (دومین کنترلر مرجع) است. علاوه بر این، از الگوریتم تبرید شبیه سازی شده [۱۶] برای حل مسئله در شبکه های با مقیاس بزرگ استفاده گردید. در [۱۷] با کمک الگوریتم خوشه بندی، شبکه به چندین زیر شبکه تجزیه شد. در این الگوریتم خوشه بندی بر اساس تراکم سوئیچها انجام می شود. سوئیچها درون یک زیر شبکه تنها با سوئیچهای همان زیر شبکه ارتباط دارند و کمتر با سوئیچهای زیر شبکه دیگر در ارتباط هستند. در هر زیر شبکه تنها یک کنترلر قرار می گیرد. ظرفیت هر کنترلر نقش اصلی در اندازه زیر شبکه دارد. در مقاله [۱۸]، مسئله مکان یابی کنترلر باهدف رسیدن به قابلیت اطمینان بین کنترلرها و سوئیچها در نظر گرفته شد. همچنین، به منظور حفاظت از کنترلرها و مسیره های کنترلی در برابر خرابیها، از یک شیوه پشتیبان گیری بر اساس مدل شکست ناپذیری نمایی استفاده گردید. مقاله [۱۹] به مطالعه مسئله مکان یابی پویا کنترلر پرداخته است. این مسئله شامل تعیین مکان ماژولهای کنترلر با محدودیت تأخیرهای ارتباطی و تعیین تعداد کنترلر در هر ماژول برای پشتیبانی از بارهای پویا است. به منظور حل این مسئله الگوریتمی به نام LiDy+ پیشنهاد گردید. این الگوریتم ترکیبی از الگوریتم مکان یابی ماژول کنترلر و الگوریتم مدیریت جریان پویا است. اخیراً، در مقاله [۲۰] مدل بیان شده در [۲۱] توسعه داده شد. در مقاله [۲۰]، در مدل شرایطی در نظر گرفته شد که در صورت اضافه شدن فهرستی از سوئیچها به شبکه فعلی هزینه کلی شبکه بعد از اضافه شدن سوئیچها دوباره حداقل گردد. به منظور حداقل کردن هزینه شبکه، مسئله مکان یابی کنترلر به صورت مجموع چند تابع هدف بیان گردید. همچنین، استدلال شد که مکان یابی کنترلر، می بایست محدودیت های خاصی را به ویژه برای سطح کنترل برآورده کند. از این رو، به طور هم زمان تعداد بهینه، مکان و نوع کنترلر و همچنین ارتباطات بین عناصر شبکه تعیین شده و برای حداقل کردن هزینه شبکه، محدودیت های مختلف در نظر گرفته شد. مدل برنامه ریزی ریاضی مطرح شده در [۲۱] و حل دقیق آن که مبتنی بر



استفاده از حل‌کننده‌های استاندارد مسائل برنامه‌ریزی ریاضی است، معیار مقایسه‌های انجام‌شده در این تحقیق خواهد بود.

### ۳- بیان دقیق مسئله به کمک مدل‌بندی ریاضی

در این بخش مدل ریاضی مسئله مکان‌یابی کنترلر، برگرفته از [۲۱]، به‌منظور بیان دقیق مسئله شرح داده می‌شود. این مدل شامل مجموعه سوئیچ‌ها، مجموعه کنترلرها، مجموعه لینک‌ها و مجموعه مکان‌های ممکن برای نصب کنترلر است که به ترتیب با نمادهای  $S$ ،  $C$ ،  $L$  و  $P$  نمایش داده می‌شود.

در این مدل، شبکه به شکل یک گراف فرض شده است. هر گره از گراف نشان‌دهنده مکان قرار گرفتن یک سوئیچ و یا کنترلر است. در حالت خاص، در یک گره سوئیچ و کنترلر می‌توانند کنار هم قرار گیرند. هر سوئیچ تنها به یکی از کنترلرها متصل است. با در نظر گرفتن محدودیت تعداد درگاه کنترلر، چندین سوئیچ می‌تواند توسط یک کنترلر مدیریت شود. همچنین در این گراف هر کنترلر می‌بایست با سایر کنترلرها در ارتباط مستقیم قرار گیرد. از این‌رو، توپولوژی القاشده به کنترلرها که در این مسئله از نوع مش کامل است.

برای هر سوئیچ  $s \in S$ ، تعداد بسته‌های ارسال‌شده به کنترلر با پارامتر  $\sigma^s$  نشان داده می‌شود. هر کنترلر  $c \in C$ ، دارای پارامترهای  $\mu^c$ ،  $\alpha^c$  و  $k^c$  است که به ترتیب بیانگر تعداد پورت، تعداد بسته، هزینه و تعداد کنترلرهای از نوع  $c$  است. لینک  $l \in L$  شامل دو پارامتر  $\omega^l$  و  $\Phi^l$  است که پهنای باند و هزینه نصب آن را بیان می‌کند. سایر پارامترهای مثبت در مدل ریاضی مسئله، مقادیر اندازه بسته، تأخیر شبکه و زمان پردازش بسته در کنترلر است که به ترتیب با نمادهای  $\beta$ ،  $\gamma$  و  $\delta$  نشان داده می‌شود. اکنون با توجه به تعریف مجموعه‌ها و پارامترها، متغیرهای تصمیم‌گیری مسئله به شرح زیر است:

$$x_{cp} = \begin{cases} 1 \\ 0 \end{cases}, \quad \text{اگر کنترلر } c \text{ در مکان } p \text{ نصب شده است،} \\ \text{در غیر این صورت،}$$

$$v_{sp}^l = \begin{cases} 1 \\ 0 \end{cases}, \quad \text{اگر لینک } l \text{ بین سوئیچ } s \text{ و کنترلر در مکان } p \text{ برقرار باشد،}$$



در غیر این صورت،

اگر لینک  $l$  بین مکان  $p$  و مکان  $q$  برقرار باشد،  

$$z_{pq}^l = \begin{cases} 1 \\ 0 \end{cases}$$
 در غیر این صورت،

هدف در این مسئله حداقل کردن هزینه پیاده‌سازی شبکه است. این هزینه‌ها شامل هزینه‌های نصب کنترلر در مکان‌ها، هزینه‌های اتصال سوئیچ‌ها به کنترلرها و نیز هزینه‌های اتصال کنترلرها به یکدیگر است که به ترتیب با نمادهای  $C_c(x)$ ،  $C_l(v)$  و  $C_t(z)$  نشان داده می‌شوند. معادلات (۱)، (۲) و (۳) نیز نحوه محاسبه هر یک از این هزینه‌ها را بیان می‌کند. تابع  $dist(a,b)$  در معادلات (۲) و (۳) فاصله اقلیدسی بین مکان  $a$  و مکان  $b$  را به دست می‌دهد.

$$C_c(x) = \sum_{c \in C} k^c \sum_{p \in P} x_{cp} \quad (۱)$$

$$C_l(v) = \sum_{l \in L} \Phi^l \sum_{s \in S} \sum_{p \in P} dist(s,p) v_{sp}^l \quad (۲)$$

$$C_t(z) = \sum_{l \in L} \Phi^l \sum_{\substack{q \in P \\ q < p}} \sum_{p \in P} dist(p,q) z_{pq}^l \quad (۳)$$

در ادامه، تابع هدف مسئله به همراه قیدهای آن بیان می‌شوند.

$$\text{Minimize } (C_c(x) + C_l(v) + C_t(z)) \quad (۴)$$

$$\sum_{c \in C} x_{cp} \leq 1, \quad \forall p \in P \quad (۵)$$

$$\sum_{q \in P} \sum_{l \in L} (z_{pq}^l + z_{qp}^l) + \sum_{s \in S} \sum_{l \in L} v_{sp}^l \leq \sum_{c \in C} \alpha^c x_{cp}, \quad \forall p \in P \quad (۶)$$

$$\sum_{l \in L} \sum_{p \in P} v_{sp}^l = 1, \quad \forall s \in S \quad (۷)$$

$$\sum_{l \in L} \sum_{s \in S} \sigma^s v_{sp}^l \leq \sum_{c \in C} \mu^c x_{cp}, \quad \forall p \in P \quad (۸)$$



$$\sum_{p \in P} x_{cp} \leq \varphi^c, \quad \forall c \in C \quad (۹)$$

$$g(\sigma^s, \beta) \leq \sum_{l \in L} f(w^l) v_{sp}^l, \quad \forall s \in S, \quad \forall p \in P \quad (۱۰)$$

$$\forall Tran(v) + \forall Prop(x, v) + Proc(x) \leq \gamma \quad (۱۱)$$

$$\sum_{c \in C} x_{cq} + \sum_{c \in C} x_{cp} \leq \sum_{l \in L} z_{pq}^l + 1, \quad (q < p, \forall q \in P, \forall p \in P) \quad (۱۲)$$

$$x_{cp} \in \{0, 1\} \quad \forall c \in C, \quad \forall p \in P \quad (۱۳)$$

$$v_{sp}^l \in \{0, 1\} \quad \forall l \in L, \quad \forall s \in S, \quad \forall p \in P \quad (۱۴)$$

$$z_{pq}^l \in \{0, 1\} \quad \forall l \in L, \quad \forall p \in P, \quad \forall q \in P \quad (۱۵)$$

قیدهای (۵) و (۶) محدودیت تعداد استفاده از کنترلر  $c$  در مکان  $p$  و محدودیت تعداد درگاه کنترلر  $c$  را بیان می‌کنند. قید (۷) بیانگر محدودیت تعداد استفاده از لینک  $l$  بین سوئیچ  $s$  و کنترلر  $c$  است. قیدهای (۸) و (۹) نشان‌دهنده محدودیت تعداد پردازش بسته توسط کنترلر  $c$  و محدودیت تعداد موجودی کنترلر  $c$  است. قید (۱۰) بیانگر محدودیت پهنای باند لینک  $l$  است. تابع  $g(a, b)$  در قید (۱۰) که دارای دو آرگومان ورودی  $a$  و  $b$  است، تعداد  $a$  بسته به طول  $b$  بایت را برحسب بایت در هر ثانیه به دست می‌آورد. تابع  $f(a)$  نیز که مقدار  $a$  در آن برحسب مگابایت یا گیگابایت است را برحسب بایت محاسبه می‌کند. قید (۱۱) و (۱۲) نشان‌دهنده محدودیت تأخیر کلی شبکه است؛ یعنی مجموع تأخیر انتقال توسط سوئیچ  $s$ ، تأخیر انتشار بین کنترلر  $c$  و سوئیچ  $s$  و تأخیر پردازش کنترلر  $c$  که به ترتیب با نمادهای  $Tran(v), Prop(x, v), Proc(x)$  نشان داده شده است، می‌بایست از تأخیر کلی شبکه کمتر یا مساوی باشد. قید (۱۲) محدودیت توپولوژی شبکه را نشان می‌دهد؛ به این معنی که توپولوژی بین گره‌های شبکه می‌بایست از نوع مش کامل باشد. در نهایت قیدهای (۱۳)، (۱۴) و (۱۵) نشان‌دهنده مقادیر باینری است که متغیرهای تصمیم‌گیری می‌توانند داشته باشند. هر تخصیص باینری به متغیرهای مسئله که قیدهای (۵) تا (۱۲) را برآورد یک جواب شدنی مسئله، به‌اختصار جواب مسئله، نامیده می‌شود.



#### ۴- الگوریتم مبتنی بر تخصیص پویای سوئیچ

در این بخش، یک الگوریتم تخصیص پویای سوئیچ (DSA)° برای حل مسئله مکان‌یابی کنترلر بکار گرفته می‌شود. این الگوریتم می‌تواند از بین کنترلرهای فعال بهترین‌ها را انتخاب کرده و سوئیچ‌ها را به آن‌ها تخصیص دهد و سایر کنترلرها را در صورت لازم خاموش کند.

هدف الگوریتم تخصیص پویای سوئیچ، کاهش تعداد کنترلرهای فعال است که منجر به کاهش هزینه در کل شبکه خواهد شد. از منظر تئوری، یک جواب بهینه برای مسئله مکان‌یابی کنترلر می‌تواند تنها با جستجو در تمام  $2^C$  زیرمجموعه تشکیل‌شده از کنترلرها به دست آید. این پیچیدگی محاسباتی برای شبکه‌های واقعی غیرعملی است. از این‌رو، نیاز است تا طراحی الگوریتم به صورتی انجام پذیرد که پیچیدگی محاسباتی برای حل مسئله مکان‌یابی کنترلر کاهش یابد. الگوریتم تخصیص پویای سوئیچ، کنترلرهای فعال را از روی ترتیب صعودی تعداد سوئیچ‌های متصل به آن‌ها بررسی و خاموش شدن آن‌ها را ارزیابی می‌کند.

یک کنترلر فعال زمانی می‌تواند خاموش شود که :

(۱) سوئیچ‌های متصل به آن بتوانند به سایر کنترلرهای فعال منتقل شوند.

(۲) هزینه شبکه با انتقال سوئیچ‌ها کاهش یابد.

دلیل انتخاب کنترلرها به ترتیب صعودی تعداد سوئیچ‌های متصل به آن‌ها این است که کنترلرهای دارای تعداد کم سوئیچ بتوانند با انتقال سوئیچ‌های خود به سایر کنترلرهای فعال خاموش شوند.

در الگوریتم تخصیص پویای سوئیچ، ابتدا برای تمام عناصر مجموعه کنترلر  $C$ ، یک عدد تصادفی بین ۰ تا ۱ تولید می‌شود که با علامت  $r$  نشان داده می‌شود. سپس کنترلرهایی که دارای مقدار  $r$  بیشتر از حد آستانه  $\theta$  باشند، به عنوان کنترلر فعال در نظر گرفته می‌شوند (در این الگوریتم مقدار حد آستانه  $\theta$  برابر  $0/8$  در نظر گرفته شده است). در ادامه، سوئیچ‌ها بر اساس فاصله به نزدیک‌ترین کنترلر فعال تخصیص می‌یابند. سپس فرآیند بررسی کنترلرهای فعال برای خاموش شدن انجام می‌پذیرد. جزئیات الگوریتم تخصیص پویای سوئیچ در الگوریتم ۱ نشان داده شده است.




**Algorithm ۱** Dynamic Switch Allocation (DSA) Algorithm

**Input:**  $C, S, x_{cp}, v_{sp}^l, z_{pq}^l \forall c \in C, \forall l \in L, \forall s \in S, \forall p \in P, \forall q \in P$   
 $k^c, \alpha^c, \mu^c, \varphi^c, \Phi^l, \omega^l, \beta, \gamma, \delta$

**Output:**  $x_{cp}^*, v_{sp}^{l*}, z_{pq}^{l*} \forall c \in C, \forall l \in L, \forall s \in S, \forall p \in P, \forall q \in P$

**۱:** **for**  $k=1 \dots 1000$  **do**

**۲:** *Compute*  $C'$ ; // Select active controllers between controllers based on  $r >$   
 $\cdot/\wedge$

**۳:**  $Best\_Cost^* = \infty, C'' = C'$ ;

**۴:** **while**  $C'' \neq 0$  **do**

**۵:** *Compute*  $x_{cp}, v_{sp}^l, z_{pq}^l \forall c \in C'', \forall l \in L, \forall s \in S, \forall p \in P,$   
 $\forall q \in P$ ; // allocation switches with controllers based on  $dist(a,b)$

**۶:** *Select*  $c^* = \arg \min \sum_{l \in L} \sum_{s \in S} x_{cp} v_{sp}^l, \forall c \in C'', \forall p \in P$ ;

**۷:**  $C'' = C'' - \{c^*\}$ ;

**۸:** **if** switches connected to  $c^*$  can be moved to other controllers **then**

**۹:**  $C' = C' - \{c^*\}$ ;

**۱۰:** *Compute*  $x_{cp}, v_{sp}^l, z_{pq}^l \forall c \in C'', \forall l \in L, \forall s \in S,$   
 $\forall p \in P, \forall q \in P$ ;

**۱۱:** *Compute*  $Best\_Cost$ ;

**۱۲:** **if**  $Best\_Cost < Best\_Cost^*$  **then**

**۱۳:**  $Best\_Cost^* = Best\_Cost, x_{cp}^* = x_{cp}, v_{sp}^{l*} = v_{sp}^l,$   
 $z_{pq}^{l*} = z_{pq}^l$ ;

**۱۴:** **else**

**۱۵:**  $C' = C' + \{c^*\}$ ;

**۱۶:** **end if**

**۱۷:** **end if**

**۱۸:** **end while**

**۱۹:** **end for**



در هر حلقه از خط ۱ الی ۱۹، در ابتدا، کنترلرهای فعال به‌طور تصادفی از میان مجموعه کنترلرهای  $C$  بر اساس مقدار  $r$  که به آن‌ها تخصیص داده می‌شود، انتخاب می‌شوند. کنترلرهایی که در آن‌ها  $r > \theta$  به‌عنوان کنترلر فعال انتخاب می‌شود. در ادامه، در طول خطوط ۴ الی ۱۸، سوئیچ‌ها به کنترلرهای فعال بر اساس نزدیک‌ترین فاصله تخصیص می‌یابند. سپس کنترلرهای فعال با کمترین تعداد سوئیچ‌های متصل انتخاب می‌شود. برای تمام سوئیچ‌های متصل به این کنترلر، امکان خاموش شدن و باز تخصیص سوئیچ‌های جاری متصل به آن، به کنترلرهای دیگر بررسی می‌شوند. برای بررسی هر کنترلر (خطوط ۹ الی ۱۶)، فرض می‌شود که این کنترلر از مجموعه کنترلرهای فعال حذف می‌شود و سوئیچ‌های آن به کنترلرهای فعال دیگر بر اساس نزدیک‌ترین فاصله باز تخصیص می‌یابد. اگر هزینه کلی شبکه کاهش یابد، پیکربندی یافته‌شده ذخیره و به‌عنوان جواب جاری در نظر گرفته می‌شود. این فرآیند تا بررسی تمام کنترلرهای فعال ادامه می‌یابد.

## ۵- نتایج شبیه‌سازی

در این بخش، آزمایش‌ها و محاسبه‌های انجام‌شده به‌منظور ارزیابی روش‌های حل مسئله مکان‌یابی کنترلر گردآوری شده است. در این راستا، کمیت‌های زمان اجرا و کیفیت جواب برای هر یک از روش‌های حل مسئله با استفاده از CPLEX [۲۲] [۲۳] [۲۴] و الگوریتم تخصیص پویای سوئیچ بر روی مجموعه‌ای از نمونه‌های استاندارد مسئله، محاسبه و در مقایسه با یکدیگر گزارش شده است.

نحوه ساخت نمونه‌های مورد آزمایش به شرح زیر است:

برای هر نمونه ابتدا توپولوژی شبکه از روی یک شبکه گرید  $20 \times 20$  و به‌طور تصادفی استخراج می‌شود. به این معنی که هر گره از گرید مذکور با احتمال  $p_r$  یک نود از گراف شبکه خواهد بود. مقدار  $p_r$  برای نمونه‌ها  $0/25$  در نظر گرفته شده است. پس از انتخاب گره‌های مذکور، گراف کامل القاء شده به آن توپولوژی شبکه از نمونه مذکور خواهد بود. روی این گراف وزن هر یال فاصله اقلیدسی بین دو سر آن تعریف می‌شود. در مرحله بعد گره‌های حاوی سوئیچ روی گراف مذکور مشخص می‌شوند. به این ترتیب که برای یک نمونه با  $i$  سوئیچ،  $i$  نود از گراف مذکور به تصادف انتخاب شده و روی هر یک از آن‌ها یک سوئیچ نصب می‌گردد.



پارامتر  $n$  مقدار سوئیچ‌های نصب‌شده در شبکه، در ساخت نمونه‌های فوق، عددی از مجموعه  $\{0, 1, \dots, 20\}$  انتخاب شده است.

محاسبات این بخش بر روی یک پردازشگر از نوع Intel Core i5 در حالت تک پردازنده و تحت سیستم عامل اوونتو با حافظه اصلی ۴ گیگابایت انجام شده است. الگوریتم تخصیص پویای سوئیچ توسط زبان C پیاده‌سازی شده است. برای CPLEX، محدودیت زمانی برابر ۳۶۰۰ ثانیه در نظر گرفته شده است. همچنین به دلیل انتخاب تصادفی کنترلرهای فعال در الگوریتم تخصیص پویای سوئیچ، برای رسیدن به جواب بهینه این الگوریتم به تعداد ۱۰ بار اجرا و میانگین آن به عنوان جواب بهینه نهایی در نظر گرفته می‌شود. پارامترهای تعریف‌شده برای حل مسئله در CPLEX و الگوریتم پیشنهادی در جدول ۱ بیان شده است.

جدول ۱ پارامترهای مسئله

مقدار	نام پارامتر
۲۵۰۰ دلار	هزینه هر کنترلر
۳۲	تعداد پورت هر کنترلر
۴۰۰۰	تعداد بسته قابل‌پردازش توسط کنترلر
۱۵	تعداد کنترلر موجود
۸/۲۵ دلار	هزینه لینک به متر
۱۰۰ مگابیت بر ثانیه	پهنای باند لینک
۱۵۰ بایت	اندازه بسته
۲۵۰ میلی‌ثانیه	حداکثر تأخیر شبکه
۰/۰۰۱ میلی‌ثانیه	متوسط زمان پردازش بسته توسط کنترلر

نتایج به‌دست‌آمده از این آزمایش‌ها در جدول ۲ بر اساس نمونه‌ها گردآوری شده‌اند. در این جدول، برای هر نمونه از مسائل موارد زیر گزارش شده‌اند.

*cost*: بهترین هزینه روی این نمونه‌ها

*time*: زمان صرف شده روی این نمونه‌ها

*imp*: درصد بهبود الگوریتم پیشنهادی DSA نسبت به CPLEX از لحاظ بهترین هزینه

روی این نمونه‌ها که به شیوه زیر محاسبه می‌شوند.



$$\frac{cost_{CPLEX}(\rho) - cost_{DSA}(\rho)}{cost_{DSA}(\rho)} \times 100 \quad (16)$$

در محاسبه این کمیت هر یک از توابع  $cost_{CPLEX}(\cdot)$  و  $cost_{DSA}(\cdot)$  به ترتیب بهترین هزینه به دست آمده از اجرای الگوریتم پیشنهادی DSA و حل کننده CPLEX را به دست می دهند.

جدول ۲ نتایج حاصل از اجرای الگوریتم DSA و CPLEX بر روی نمونه ها

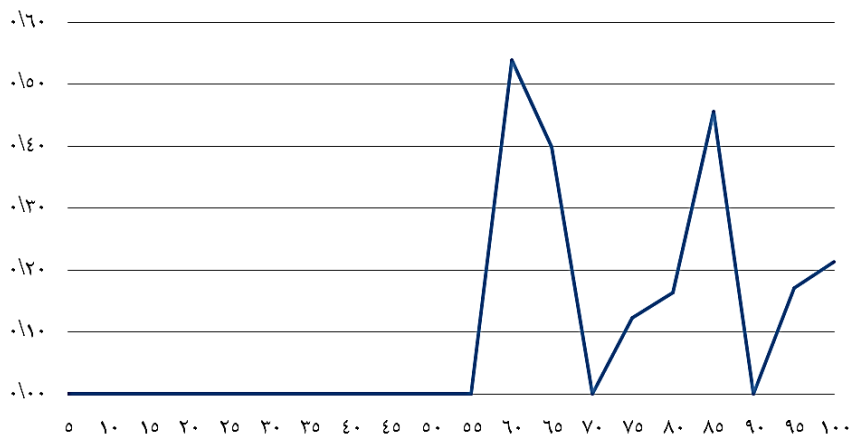
instance switch	CPLEX		DSA		CPLEX vs DSA
	time (sec.)	cost \$	time (sec.)	Cost \$	Imp %
۵	۴/۶۲	۲۴۴۴۹۵	۷/۹	۲۴۴۴۹۵	-/۰۰
۱۰	۱۶/۱۹	۴۵۷۳۲۵	۸/۰۳	۴۵۷۳۲۵	-/۰۰
۱۵	۱۸/۴۹	۵۷۷۶۱۶	۹/۲۹	۵۷۷۶۱۶	-/۰۰
۲۰	۳۵۶/۵۱	۹۶۸۵۴۷	۱۰/۵۱	۹۶۸۵۴۷	-/۰۰
۲۵	۱۷۹/۶۹	۱۲۰۳۸۷۸	۹/۴۹	۱۲۰۳۸۷۸	-/۰۰
۳۰	۲۵۱/۵۹	۱۱۹۵۴۳۰	۵/۳۶	۱۱۹۵۴۳۰	-/۰۰
۳۵	۱۵۱۶/۶۲	۱۶۰۷۴۰۷	۱۲/۳۹	۱۶۰۷۴۰۷	-/۰۰
۴۰	۷۲۵/۵۸	۱۷۰۷۱۱۰	۱۳/۷۵	۱۷۰۷۱۱۰	-/۰۰
۴۵	۶۴۲/۶۲	۱۷۶۶۷۲۰	۱۲/۸۲	۱۷۶۶۷۲۰	-/۰۰
۵۰	۱۲۷۷/۳۷	۲۰۴۸۱۱۳	۱۴/۹۲	۲۰۴۸۱۱۳	-/۰۰
۵۵	۱۹۷۸/۱۳	۲۱۳۷۹۶۵	۱۵/۱۲	۲۱۳۷۹۶۵	-/۰۰
۶۰	>۳۶۰۰	۲۲۹۱۰۷۵	۱۵/۷۲	۲۲۷۸۸۰۸	-/۵۴
۶۵	>۳۶۰۰	۲۴۳۱۴۵۱	۱۶/۶۷	۲۴۲۱۷۸۴	-/۴۰
۷۰	>۳۶۰۰	۲۵۵۵۷۹۰	۱۷/۱۱	۲۵۵۵۷۹۰	-/۰۰
۷۵	>۳۶۰۰	۲۷۶۹۹۱۷	۱۸/۰۸	۲۷۶۶۵۲۷	-/۱۲
۸۰	>۳۶۰۰	۲۸۷۰۷۳۷	۱۹/۰۲	۲۸۶۶۰۵۶	-/۱۶
۸۵	>۳۶۰۰	۳۰۳۵۱۸۷	۱۹/۴۴	۳۰۲۱۴۲۴	-/۴۶
۹۰	>۳۶۰۰	۳۱۹۰۶۹۴	۲۰/۳۱	۳۱۹۰۶۹۴	-/۰۰
۹۵	>۳۶۰۰	۳۳۶۳۸۷۵	۲۰/۹۹	۳۳۵۸۱۱۴	-/۱۷
۱۰۰	>۳۶۰۰	۳۴۶۱۵۹۶	۲۱/۷۷	۳۴۵۴۲۲۱	-/۲۱



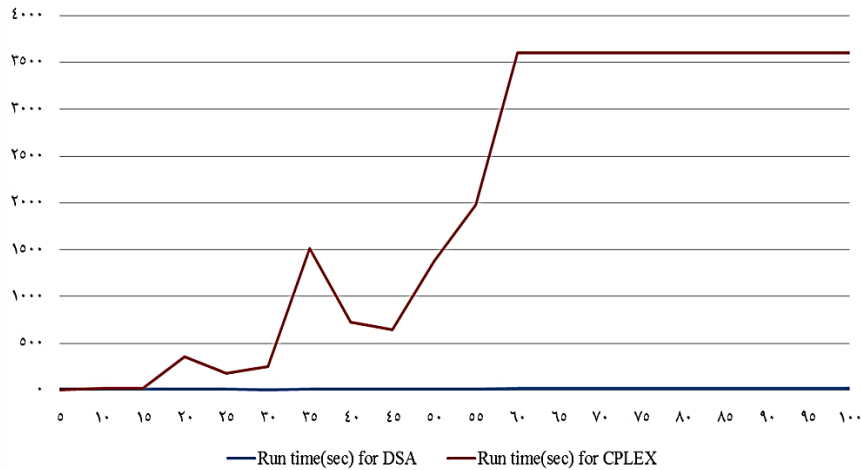
در جدول ۲، ستون اول نام نمونه‌های مورد آزمایش را بیان می‌کند. ستون دوم، مدت‌زمان اجرا CPLEX بر روی نمونه‌ها را نشان می‌دهد. ستون سوم نشان‌دهنده مقدار هزینه بهترین جواب به‌دست‌آمده برای هر نمونه توسط CPLEX با در نظر گرفتن حداکثر مدت‌زمان اجرا ۳۶۰۰ ثانیه است. ستون چهارم و پنجم نیز به ترتیب نشان‌دهنده مدت‌زمان اجرا و مقدار هزینه بهترین جواب به‌دست‌آمده توسط الگوریتم پیشنهادی DSA است. در انتها، ستون ششم درصد بهبودی الگوریتم پیشنهادی DSA در مقایسه با CPLEX از لحاظ مقدار هزینه بهترین راه‌حل به‌دست‌آمده را بیان می‌کند.

نتایج به‌دست‌آمده از جدول ۲ به برتری الگوریتم پیشنهادی DSA در زمان اجرا برای تمام نمونه‌ها و بهترین هزینه به‌دست‌آمده در برخی از این نمونه‌ها اشاره دارد.

شکل ۱ درصد بهبودی الگوریتم پیشنهادی DSA در مقایسه با CPLEX برای نمونه‌های مختلف را نشان می‌دهد. محور افقی اندازه نمونه‌ها بر اساس تعداد سوئیچ نصب‌شده در شبکه و محور عمودی درصد بهبودی روی نمونه را نشان می‌دهد. با توجه به این نمودار، برتری الگوریتم DSA در یافتن جواب‌هایی با هزینه کمتر با افزایش تعداد سوئیچ‌های موجود در نمونه و در مقایسه با CPLEX رفته‌رفته بیشتر می‌شود. این مهم، الگوریتم پیشنهادی را به‌عنوان ابزاری مهم برای حل مسئله در مقیاس بزرگ مطرح می‌کند.



شکل ۱ درصد بهبود به‌دست‌آمده از اجرای الگوریتم DSA در مقایسه با CPLEX



شکل ۲ زمان اجرای الگوریتم DSA روی نمونه‌های هم‌اندازه در مقایسه با CPLEX

شکل ۲، زمان اجرای الگوریتم DSA را در مقایسه با CPLEX در نظر قرار داده است. محور افقی در این شکل، اندازه نمونه‌ها بر اساس تعداد سوئیچ نصب‌شده و محور عمودی زمان اجرا روی نمونه‌ها را نشان می‌دهد. نتایج به‌دست‌آمده در این شکل، به برتری زمانی روش پیشنهادی اشاره دارد. به‌بیان‌دیگر، الگوریتم پیشنهادی قادر است با به‌کارگیری مناسب منبع زمان، جواب‌های بهتری را در زمان کوتاه‌تر بیابد. این مهم، روش پیشنهادی را به‌عنوان روش کارا در حل مسئله مکان‌یابی کنترلر مطرح خواهد کرد.

## ۶- نتیجه‌گیری

شبکه‌های مبتنی بر نرم‌افزار، نسل جدیدی از شبکه‌های کامپیوتری است که با جداسازی سطح داده از سطح کنترل مزایای بسیاری در مقایسه با شبکه‌های سنتی به ارمغان آورده است. یکی مسئله‌های مطرح در این شبکه‌ها مکان قرارگیری کنترلر است که در این مقاله مورد مطالعه قرار گرفته است. برای حل این مسئله الگوریتمی مبتنی بر روش تخصیص پویای سوئیچ پیشنهاد شده است. الگوریتم پیشنهاد شده از روش خاموش کردن کنترلرهای فعال برای رسیدن



به هدف استفاده می‌کند تا به توپولوژی کارآمدی از شبکه در یک‌زمان قابل قبول دست یابد. به‌طوری‌که حداقل تعداد مورد نیاز کنترلرها برای تخصیص سوئیچ‌ها استفاده شود و هزینه پیاده‌سازی شبکه کاهش یابد. به‌منظور بررسی عملکرد الگوریتم پیشنهادی از لحاظ زمان اجرا و یافتن بهترین جواب بهینه، آزمایش‌هایی بر روی چندین نمونه از شبکه انجام شده است. این نمونه شبکه‌ها به‌طور تصادفی ایجاد می‌گردند. نتایج به‌دست‌آمده از الگوریتم پیشنهادی و نتایج CPLEX که بر روی همان نمونه‌ها اجرا شده است، مورد مقایسه قرار گرفته است. نتایج مقایسه، از برتری الگوریتم پیشنهادی در زمان اجرا برای تمام نمونه‌ها و در بهترین هزینه به‌دست‌آمده در برخی از نمونه‌ها حکایت دارد. علاوه بر این، می‌توان نتیجه گرفت که الگوریتم پیشنهادی در صورت بزرگ‌تر شدن اندازه شبکه دارای برتری قابل مشهودی هم از لحاظ زمان اجرا و هم بهترین هزینه خواهد بود. از این‌رو، به‌کارگیری الگوریتم‌های ابتکاری می‌تواند در یافتن بهترین هزینه در کوتاه‌ترین زمان در مقایسه با ابزارهای مانند CPLEX مؤثرتر باشد. در کارهای آتی سعی خواهیم کرد پویایی شبکه را در زمانی که بسته‌ها با اندازه‌های متغیر در طول شبکه ارسال خواهند شد، بررسی کنیم. همچنین، حالت‌های مختلف خرابی شبکه را در کارمان در نظر خواهیم گرفت. علاوه بر این، به‌کارگیری از سایر الگوریتم‌های ابتکاری جهت یافتن به بهترین جواب در کمترین زمان را مورد بررسی قرار خواهیم داد.

## ۷- پی‌نوشت‌ها

1. Software Defiend Network
2. Open Networking Foundation
3. Heuristic
4. Mixed-Integer Linear Programming
5. Dynamic Switch Allocation



## ۸- منابع

- [1] Singh, A. K., & Srivastava, S. (2018). A survey and classification of controller placement problem in SDN. *International Journal of Network Management*, 28(3), e2018.
- [2] Sood, K., & Xiang, Y. (2017). The controller placement problem or the controller selection problem?. *Journal of Communications and Information Networks*, 2(3), 1-9.
- [3] Nunes, B. A. A., Mendonca, M., Nguyen, X. N., Obraczka, K., & Turletti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications surveys & tutorials*, 16(3), 1617-1634.
- [4] Xia, W., Wen, Y., Foh, C. H., Niyato, D., & Xie, H. (2014). A survey on software-defined networking. *IEEE Communications Surveys & Tutorials*, 17(1), 27-51.
- [5] Open networking foundation.  
<https://www.opennetworking.org/about>.
- [6] Jarraya, Y., Madi, T., & Debbabi, M. (2014). A survey and a layered taxonomy of software-defined networking. *IEEE communications surveys & tutorials*, 16(4), 1955-1980.
- [7] Sezer, S., Scott-Hayward, S., Chouhan, P. K., Fraser, B., Lake, D., Finnegan, J., ... & Rao, N. (2013). Are we ready for SDN? Implementation challenges for software-defined networks. *IEEE Communications Magazine*, 51(7), 36-43.
- [8] Heller, B., Sherwood, R., & McKeown, N. (2012). The controller placement problem. *ACM SIGCOMM Computer Communication Review*, 42(4), 473-478.
- [9] Moradi, A., & Valinejad, A. (2018). Accelerated Heuristic Approaches to the Capacitated Facility Location Problem. *Modern Researches in Decision Making*, 3(3), 191-211.
- [10] BabaeiMorad, S., Bagheri, H., & Behnamian, J. (2019). A Hierarchical Covering Location Model with a Multi Period under Uncertainty. *Modern Researches in Decision Making*, 4(1), 25-53.
- [11] Yao, G., Bi, J., Li, Y., & Guo, L. (2014). On the capacitated controller placement problem in software defined networks. *IEEE Communications Letters*, 18(8), 1339-1342.
- [12] Ahmadi, V., & Khorramizadeh, M. (2018). An adaptive heuristic for multi-objective controller placement in software-defined networks. *Computers &*





- Electrical Engineering, 66, 204-228.
- [13] Wang, G., Zhao, Y., Huang, J., & Wu, Y. (2017). An effective approach to controller placement in software defined wide area networks. *IEEE Transactions on Network and Service Management*, 15(1), 344-355..
- [14] Killi, B. P. R., & Rao, S. V. (2017). Capacitated next controller placement in software defined networks. *IEEE Transactions on Network and Service Management*, 14(3), 514-527.
- [15] Azar A., Toghyani A. (2014). A Review Of Full Fuzzy Linear Programming Problems. *Management Research In Iran (Modares Human Sciences)*, 18(1), 55-82.
- [16] Bashiri M., Garmeyi Y. (2014). A Solution Approach For Multi-Criteria Gradual Covering Location Problem Using Simulated Annealing And Artificial Neural Network. *Management Research In Iran (Modares Human Sciences)*, 17(4), 25-41.
- [17] Liao, J., Sun, H., Wang, J., Qi, Q., Li, K., & Li, T. (2017). Density cluster based approach for controller placement problem in large-scale software defined networkings. *Computer Networks*, 112, 24-35.
- [18] Wang, Y., Zhong, Q., Qiu, X., & Li, W. (2018). Resource allocation for reliable communication between controllers and switches in SDN. *Journal of Network and Systems Management*, 26(4), 966-992.
- [19] ul Huque, M. T. I., Si, W., Jourjon, G., & Gramoli, V. (2017). Large-scale dynamic controller placement. *IEEE Transactions on Network and Service Management*, 14(1), 63-76.
- [20] Sallahi, A., & St-Hilaire, M. (2016). Expansion model for the controller placement problem in software defined networks. *IEEE Communications Letters*, 21(2), 274-277.
- [21] Sallahi, A., & St-Hilaire, M. (2015). Optimal model for the controller placement problem in software defined networks. *IEEE communications letters*, 19(1), 30-33.
- [22] *CPLEX: IBM's Linear Programming Solver*.
- [23] Mueller, J., Wierz, A., & Magedanz, T. (2013, November). Scalable on-demand network management module for software defined telecommunication networks. In *2013 IEEE SDN for Future Networks and Services (SDN4FNS)* (pp. 1-6). IEEE.
- [24] Herbaut, N., Negru, D., Magoni, D., & Frangoudis, P. A. (2016, July). Deploying a content delivery service function chain on an SDN-NFV operator infrastructure. In *2016 International Conference on Telecommunications and Multimedia (TEMU)* (pp. 1-7). IEEE.