مجله علمی- پژوهشی
«عمـران مـــدرس»
دوره پانزدهم، شماره ۲، تابستان ۱۳۹۴

موازی‌سازی الگوریتم کلونی مورچگان در طراحی شبکه گسسته حمل‌ونقل

امیرعلی زرین‌مهر[۱*]، مرتضی پرویزی‌عمران[۲]، یوسف شفاهی[۳]، سیداحسان سیدابریشمی[۴]

۱- کارشناس‌ارشد برنامه‌ریزی حمل‌ونقل، دانشکده مهندسی عمران، دانشگاه صنعتی‌شریف
۲- کارشناس‌ارشد مهندسی کامپیوتر-الگوریتم‌ها و محاسبات، دانشکده‌فنی، دانشگاه‌تهران
۳- استاد برنامه‌ریزی حمل‌ونقل، دانشکده مهندسی عمران، دانشگاه صنعتی‌شریف
٤- استادیار برنامه‌ریزی حمل‌ونقل، دانشکده مهندسی عمران و محیط زیست، دانشگاه تربیت‌مدرس

amirali.zarrinmehr@modares.ac.ir
تاریخ دریافت: ۹۳/۲/۳۰ تاریخ پذیرش: ۹۳/۱۰/۲۳

چکیده- طراحی شبکه گسسته حمل‌ونقل عبارت است از انتخاب زیرمجموعه‌ای امکان‌پذیر از پروژه‌ها(بزرگراه‌ها)ی پیشنهادی در یک شبکه حمل‌ونقل به منظور کمینه‌سازی زمان سفر کل کاربران شبکه. این مساله در کلاس پیچیدگی مسائل NP-Hard قرار دارد که هیچ الگوریتم موثری برای حل دقیق آن‌ها در مقیاس بزرگ وجود ندارد. ازاین‌رو بیشتر مطالعات انجام‌گرفته، به منظور یافتن جواب خوبی در مدت زمان معقول، از طریق رویکردهای ابتکاری و فرا ابتکاری به مساله پرداخته‌اند. اما راه دیگری که همچنان برای افزایش سرعت رویکردهای حل مساله وجود دارد، محاسبات موازی است. مقاله پیش‌رو، در پی بررسی کاربرد محاسبات موازی در یک الگوریتم فراابتکاری در مساله طراحی شبکه گسسته حمل‌ونقل است. در این مقاله، یک الگوریتم موازی کلونی مورچگان، بر مبنای مطالعه پورزاهدی و ابوالقاسمی، با الگوی موازی‌سازی ارباب-کارگر پیشنهاد می‌شود. برای مطالعه موردی، شبکه حمل‌ونقلی خلاصه‌شده شیکاگو با ۱٦ پروژه پیشنهادی درنظرگرفته می‌شود. نتایج موازی‌سازی روی خوشه‌ای از ۸ هسته پردازشی نشان‌دهنده آن است که الگوریتم‌های موازی می‌توانند ظرف مدت زمان ٤ هزار ثانیه به جواب‌هایی با کیفیت بالا دست پیدا کنند، درحالی‌که همین دستیابی برای الگوریتم‌های تک‌هسته‌ای در مدت ۱۰ هزار ثانیه اتفاق می‌افتد. نتایج نشان می‌دهند که در دو مورد از سه اجرا، الگوریتم موازی کلونی مورچگان به جواب دقیق مساله دست می‌یابد، و در مورد دیگر به جوابی با ۰/۰۷ درصد خطا نسبت به جواب دقیق همگرا می‌شود. عملکرد موازی الگوریتم کلونی مورچگان، همچنین در کنار الگوریتم شاخه‌وکرانه موازی گزارش خواهدشد. این گزارش نشان می‌دهد که الگوریتم موازی شاخه‌وکرانه، در مثال مورد نظر، به زمان بسیارزیادی، بیش از ۳۲ هزار ثانیه، برای یافتن جواب دقیق مساله احتیاج دارد. البته قضاوت و مقایسه کلی، در خصوص رفتار الگوریتم‌های یادشده به اجراهای بیشتر بر روی شبکه‌های گوناگون نیازمند است.

واژگان کلیدی: طراحی شبکه گسسته حمل‌ونقل، الگوریتم کلونی مورچگان، محاسبات موازی، الگوی ارباب-کارگر.

۱- مقدمه

مجموعه مسائل طراحی شبکه، شامل دسته وسیعی از مسائل زیربنایی، در حوزه‌های برنامه‌ریزی حمل‌ونقل، مخابرات، صنایع، و علوم‌نظامی است. این مسائل، به شکل معمول با یک گراف، به عنوان نمادی از شبکه، نمایش داده‌می‌شوند که دارای تعدادی گره و یال جهت‌دار یا بدون‌جهت است و هر یک از یال‌ها، خصوصیاتی همچون طول، ظرفیت، و هزینه را به همراه دارند. هدف مسائل طراحی شبکه انتخاب تعدادی پروژه (یال در گراف) به منظور ساخت یا توسعه ظرفیت است، به گونه‌ای که نتیجه خاصی همچون کیمنه‌سازی هزینه انتقال تقاضا بین مبدا-مقصدها تامین شود [۱]. مقاله پیش‌رو از میان مجموعه مسائل طراحی شبکه، بر روی طراحی شبکه گسسته حمل‌ونقل به عنوان یک مساله تصمیم‌گیری زیرساختی در برنامه‌ریزی حمل‌ونقل متمرکز می‌شود.

به طور خاص، مساله طراحی شبکه گسسته حمل‌ونقل را می‌توان به صورت انتخاب زیرمجموعه‌ای امکان‌پذیر از پروژه‌های پیشنهادی ساخت بزرگراه (یا به اختصار پروژه‌ها) در یک شبکه بزرگراهی حمل‌ونقل به منظور دستیابی به هدفی مشخص تعریف کرد. این هدف به طور معمول کمینه‌سازی کل زمان سفر کاربران از شبکه است [۲].

اغلب مسائل طراحی شبکه، از جمله مساله طراحی شبکه گسسته حمل‌ونقل، مسائلی ترکیبی در رده پیچیدگی NP-Hard هستند، که دستیابی به جواب دقیق آنها در نمونه‌های بزرگ ناممکن است [۳و٤و٥]. برای حل این‌گونه مسائل ترکیبی در مقیاس واقعی دو رویکرد اساسی می‌تواند مورد توجه قرار گیرد: استفاده از الگوریتم‌های ابتکاری-فراابتکاری، و بهره‌گیری از محاسبات موازی [٦]. اگرچه در برنامه‌ریزی حمل‌ونقل تاکنون پژوهش‌های فراوانی با استفاده از الگوریتم‌های ابتکاری-فراابتکاری به مساله طراحی شبکه گسسته حمل‌ونقل پرداخته‌اند، اما کاربرد محاسبات موازی در حل این مساله بسیار محدود بوده‌است.

این مطالعه، با توجه به ضرورت حل مسائل بزرگ طراحی شبکه، از یک سو، و کاربرد موفق محاسبات موازی در حل مسائل زمان‌بر، از سوی دیگر، به دنبال بررسی اثر موازی‌سازی در حل مساله یادشده از طریق الگوریتم کلونی مورچگان است. گزارش نتایج موازی‌سازی و همچنین مقایسه آن‌ها با حل دقیق، در این مطالعه، نشان‌دهنده آن است که الگوریتم موازی کلونی مورچگان می‌تواند با دقت بالایی در مدت زمانی به مراتب کمتر از الگوریتم‌های غیرموازی به جواب مساله دست پیدا کند.

۲- بیان رسمی مساله

تعریف مساله طراحی شبکه گسسته حمل‌ونقل به شکل معمول در قالب یک مدل دوسطحی انجام می‌شود؛ مدلی که در سطح پایین، سعی در ملاحظه رفتار کاربران در انتخاب مسیر داشته و در سطح بالایی، مجموعه پروژه‌های بهینه را از میان پروژه‌های پیشنهادی انتخاب می‌کند [۷]. به منظور تعریف رسمی مساله، در سطح بالا فرض می‌شود:

A= مجموعه یال(بزرگراه)های جهت‌دار شبکه،

A_y = مجموعه پروژه(بزرگراه)های پیشنهادی برای ساخت،

y_a = متغیر تصمیم، با مقادیر ۱ یا ۰ به‌ترتیب به‌معنای انتخاب یا عدم انتخاب پروژه a برای ساخت، $a \in A_y$،

c_a = هزینه ساخت پروژه a، $a \in A_y$،

y = بردار متغیرهای تصمیم y_a،

$A(y)$ = مجموعه یال‌های شبکه با در نظرگیری بردار y،

x_a = جریان ترافیک تخصیص‌یافته به یال a، از حل مساله تخصیص ترافیک، $a \in A(y)$،

$x(y)$ = بردار مقادیر جریان تخصیص‌یافته به یال‌های شبکه پس از درنظرگیری بردار y،

$t_a(x_a)$ = تابع زمان سفر–حجم برای یال a که به صورت صعودی، محدب و مشتق‌پذیر فرض می‌شود، $a \in A(y)$،

B = مقدار بودجه دردست برای ساخت پروژه‌های پیشنهادی.

برمبنای تعاریف انجام‌گرفته، شکل رایج سطح بالایی مدل مساله را می‌توان در روابط (۱) تا (٤) مشاهده کرد [۲و۸]:

$$\text{Min}_y \ \sum_{a \in A(y)} x_a t_a(x_a) \quad (۱)$$

$$\text{s.t} \ . \sum_{a \in A_y} c_a y_a \le B \quad (۲)$$

$$y_a = 0 \text{ یا } 1 \ \ \forall a \in A_y \quad (۳)$$

$x(y)$ = جواب حاصل از حل مساله تخصیص ترافیک (٤)

سطح پایین این مساله (انتخاب مسیر کاربران شبکه که به صورت محدودیت (٤) در سطح بالایی مدل دیده می‌شود) نیز یک مساله کلاسیک حمل‌ونقلی موسوم به مساله تخصیص‌ترافیک است، که خود دارای مدل‌ها و الگوریتم‌های حل گوناگونی است [۹و۱۰و۱۱] که برخی از این الگوریتم‌ها با آثار جالبی در حل مساله طراحی شبکه حمل‌ونقل همراه بوده‌اند [۱۲]. آنچه در این مطالعه مورد استفاده قرار می‌گیرد، حل یک مساله برنامه‌ریزی ریاضی در قالب روابط (۵) تا (۸) است [۹]. تعاریف مورد نیاز در این روابط، از قرار زیر هستند:

P = مجموعه زوج‌های مبدا–مقصد در شبکه،

d_{rs} = تقاضای سفر از مبدا r به مقصد s، $(r,s) \in P$،

$N(y)$ = شبکه حاصل پس از درنظرگیری بردار y،

$K_{rs}(y)$ = مجموعه ناتهی مسیرهای از r به s در شبکه پس از درنظرگیری بردار y، $(r,s) \in P$،

f_k = جریان در مسیر k از مبدا r به مقصد s در شبکه پس از درنظرگیری بردار y، $k \in K_{rs}(y)$،

δ_{ak} = متغیر دودویی، ۱ یا ۰ به ترتیب در صورتی که یال a در مسیر k واقع باشد یا نباشد.

$$\text{Min}_{x(y)} \ \sum_{a \in A(y)} \int_0^{x_a} t_a(u)\,du \quad (۵)$$

$$\text{s.t.} \ \sum_{k \in K_{rs}(y)} f_k = d_{rs} \ \forall (r,s) \in P \quad (٦)$$

$$x_a = \sum_{(r,s) \in P} \sum_{k \in K_{rs}(y)} \delta_{ak} f_k \ \ \forall a \in A(y) \quad (۷)$$

$$f_k \ge 0 \ \ \forall k \in K_{rs}(y) \ \forall (r,s) \in P \quad (۸)$$

مساله تخصیص ترافیک با بیان بالا، دارای تابع هدف کوژ و محدودیت‌های خطی است. ازاین‌رو، روش‌های رایج در برنامه‌ریزی کوژ از جمله الگوریتم تکراری ترکیب کوژ، برای حل آن قابل کاربرد است [۱۳].

۳- مرور ادبیات موضوع

مساله طراحی شبکه حمل‌ونقل دارای ادبیات گسترده‌ای است که مراجع [۱٤،۲، و ۱۵] طبقه‌بندی مناسبی از آن ارائه نموده‌اند. این بخش، با تمرکز بر موضوع مقاله، ادبیات مرتبط را در سه دسته بررسی می‌کند. ابتدا کاربرد الگوریتم‌های فراابتکاری در طراحی شبکه گسسته حمل‌ونقل به‌اختصار مرور می‌شود. سپس برخی مفاهیم پیش‌نیاز در محاسبات موازی معرفی شده، در پایان، کاربرد محاسبات موازی در مجموعه مسائل طراحی شبکه بیان می‌شود.

۳-۱- الگوریتم‌های فراابتکاری در طراحی شبکه گسسته حمل‌ونقل

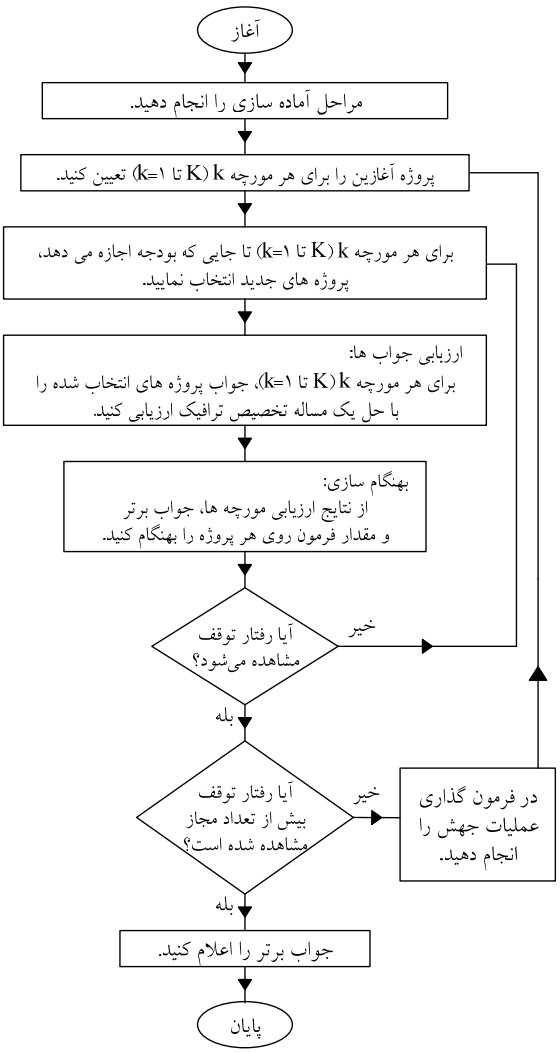
پژوهش‌های پیشین به‌شکل نظری نشان داده‌اند که مساله طراحی شبکه گسسته حمل‌ونقل، درقالب یک مساله دوسطحی، حتی در ساده‌ترین شکل آن (با فرض خطی بودن مسائل سطوح بالا و پایین) به لحاظ پیچیدگی در رده مسائل NP-Hard قرار دارد [٤]. ازاین‌رو، بیشتر مطالعات اخیر از حل دقیق این مساله چشم‌پوشی کرده، به راه‌حل‌های ابتکاری–فراابتکاری روی آوردند. الگوریتم‌های فراابتکاری، اگرچه هرگز تضمینی برای بهینگی جواب خود ارائه نمی‌دهند، اما همواره به صورت راهی برای به دست

آوردن جواب‌های نسبتا خوب در مدت زمانی معقول مطرح است [۲و۸].

تاکنون الگوریتم‌های فراابتکاری فراوانی همچون ژنتیک، جستجوی ممنوعه، گرم‌وسردکردن شبیه‌سازی شده، کلونی مورچگان، و اجتماع ذرات، در حل مساله طراحی شبکه حمل‌ونقلی به‌کارگرفته شده‌اند [۲ و ۵ و ۸ و ۱٦ و ۱۷]. الگوریتم کلونی مورچگان، ارائه‌شده به وسیله‌ی پورزاهدی و ابوالقاسمی [۲]، یکی از نمونه‌های موفق و پرارجاع در ادبیات است، که شکل (۱) فلوچارت کلی آن را نشان می‌دهد.

در الگوریتم کلونی مورچگان، ارائه شده به وسیله‌ی پورزاهدی و ابوالقاسمی [۲]، تعداد K مورچه (برابر با تعداد پروژه‌های مورد انتخاب) هر یک به‌طور تصادفی بر مبنای احتمالات حاصل از دیدپذیری و فرمون‌گذاری، زیرمجموعه‌ای امکان‌پذیر از پروژه‌ها را برگزیده و جواب به دست‌آمده را با حل یک مساله تخصیص ترافیک ارزیابی می‌کنند. سپس نتایج ارزیابی مورچگان، از طریق افزایش سطح فرمون و احتمال انتخاب در پروژه‌های دارای جواب برتر، الگوریتم را به سوی انتخاب پروژه‌های برتر (دارای فرمون و دیدپذیری بیشتر) هدایت می‌کند.

این روند، تا جایی ادامه پیدا می‌کند که در عملکرد الگوریتم رفتار توقف مشاهده شود. در این شرایط، پورزاهدی و ابوالقاسمی، با تغییر مصنوعی فرمون‌گذاری به نفع پروژه‌های ضعیف (دارای فرمون کمتر)، حرکت الگوریتم را به‌سوی فرار از توقف در جواب‌های بهینه محلی سوق می‌دهند. از این عملیات در فلوچارت شکل (۱) به عنوان جهش یاد شده‌است. عملیات جهش، خود به تعداد مشخصی (سه‌بار در مطالعه پورزاهدی و ابوالقاسمی) تکرار می‌شود تا شانس گرفتاری الگوریتم در جواب‌های بهینه محلی کمتر شود. جزییات بیشتر در خصوص این الگوریتم را می‌توان در منبع [۲] یا [۱۸] ملاحظه کرد.



شکل (۱) فلوچارت کلی برای الگوریتم کلونی مورچگان در حل مساله طراحی شبکه گسسته حمل‌ونقل [۲]

۳-۲- محاسبات موازی

به شکل کلی به استفاده هم‌زمان از چندین منبع محاسباتی در حل یک مساله، محاسبات موازی گفته می‌شود [۱۹]، که منظور از منبع محاسباتی در این تعریف، بیشتر هسته‌های پردازشی رایانه‌هاست. محاسبات موازی یا چند هسته‌ای، در مقابل محاسبات ترتیبی یا تک‌هسته‌ای، قرار می‌گیرد که به

طور سنتی رواج داشته‌است. در محاسبات تک‌هسته‌ای، عملیات محاسبات، نه به طور هم‌زمان، بلکه به ترتیب، یکی بعد از دیگری با گذشت زمان و هزینه قابل‌توجه انجام می‌گرفته است. نیاز به صرفه‌جویی در زمان و هزینه، به‌علاوه ضرورت حل مسائل محاسباتی بزرگ، موجب شد تا در دهه‌های اخیر، طراحی الگوریتم‌های موازی با اقبال گسترده‌ای روبرو شود [۱۹].

الگوهای گوناگونی برای طراحی یک الگوریتم به طور موازی وجود دارد که در یک طبقه‌بندی کلی می‌توان این الگوها را ناشی از پاسخ به دو پرسش دانست [۱۹و۲۰]: پردازنده‌ها چه هنگام با یکدیگر ارتباط برقرار می‌کنند، و پردازنده‌ها چگونه با یکدیگر ارتباط برقرار می‌کنند. الگوی ارباب-کارگر یکی از الگوهای پایه در طراحی الگوریتم‌های موازی است که در آن همه پردازنده‌ها (کارگران) در زمان‌های از پیش تعیین‌شده، با یک پردازنده مشخص (ارباب) ارتباط برقرار می‌کنند.

از نکات مهم در طراحی یک الگوریتم موازی با الگوی ارباب-کارگر، توجه به سربار ارتباطات و سربار بیکاری است، که موجب هدررفت منابع محاسباتی می‌شود [۲۰و۲۱]. منظور از سربار ارتباطات مدت‌زمانی است که به جای عملیات موازی، صرف برقراری ارتباط پردازنده‌های کارگر با پردازنده ارباب می‌شود. هرچه حجم تبادل‌داده بین پردازنده‌ها بیشتر باشد، سربار ارتباطات افزایش یافته و از کارایی الگوریتم موازی کاسته می‌شود. همچنین منظور از سربار بیکاری مدت زمان بیکاری‌است که برخی پردازنده‌های کارگر در فاصله بین دو ارتباط با پردازنده ارباب تجربه می‌کنند. برای کاهش سربار بیکاری، باید تا جای ممکن از تفاوت در حجم عملیات پردازنده‌های کارگر جلوگیری نمود [۲۰و۲۱].

۳-۳- کاربرد الگوریتم‌های موازی در مساله طراحی شبکه گسسته حمل‌ونقل

جستجو در ادبیات موضوع مجموعه مسائل طراحی شبکه حاکی از آن است که مطالعاتی که در این حوزه به موازی‌سازی الگوریتم‌های حل پرداخته‌اند، بسیار محدود بوده و بیشتر آنها مربوط به مسائل مخابرات است [۱]. در حوزه برنامه‌ریزی حمل‌ونقل، بنا به اطلاع این مطالعه، کاربرد الگوریتم‌های موازی در مساله طراحی شبکه به مطالعات اگروال و ماتیو [۲۲]، یانگ و همکاران [۲۳]، و زرین‌مهر [۲٤] محدود می‌شود. از میان این مطالعات، دو مطالعه نخست به موازی‌سازی الگوریتم‌های ابتکاری-فراابتکاری در حمل‌ونقل همگانی مربوط می‌شوند، اما مطالعه سوم، به مساله‌ای مشابه با این مقاله می‌پردازد.

در مطالعه زرین‌مهر [۲٤]، حل دقیق مساله طراحی شبکه گسسته حمل‌ونقل از طریق یک الگوریتم شاخه‌وکرانه موازی مبتنی بر مطالعه لبلانک [۲۵] مورد توجه قرار می‌گیرد. در این مطالعه، پردازنده ارباب اطلاعات اصلی الگوریتم شاخه‌وکرانه را در اختیار دارد، که عبارت‌اند از گره‌های فعال درخت جستجو (درخت شاخه‌وکرانه) و بهترین جواب جاری. در آغاز هر تکرار، پردازنده ارباب از گره‌های فعال، گره‌های جدیدی ساخته و آن‌ها را برای ارزیابی به پردازنده‌های کارگر می‌سپارد. در پایان هر تکرار نیز، کارگران نتایج ارزیابی خود را به پردازنده ارباب بازمی‌گردانند، و پردازنده ارباب، به کمک نتایج دریافتی، اطلاعات اصلی الگوریتم را بهنگام می‌کند. این فرایند آن‌قدر ادامه پیدا می‌کند که هیچ گره فعالی باقی نماند. در این وضعیت، پردازنده ارباب بهترین جواب جاری را به عنوان جواب بهینه جهانی (جواب حل دقیق) مساله گزارش می‌کند.

گزارش نتایج مطالعه زرین‌مهر بر روی ۱ الی ۱٦ هسته پردازشی نشان‌دهنده آن است که با افزایش اولیه تعداد هسته‌ها، سرعت‌حل به طور خطی افزایش می‌یابد. همچنین، اگرچه برای تعداد هسته‌های پردازشی بیشتر، افزایش سرعت حل از حالت خطی فاصله می‌گیرد، اما این افزایش همچنان مقداری قابل‌توجه است.

٤- روش حل مساله

چنان‌که در بخش ۱ (مقدمه) مطرح شد، این مطالعه در پی بررسی رفتار موازی یک الگوریتم فراابتکاری در حل مساله طراحی شبکه گسسته حمل‌ونقل است. در این راستا، الگوریتم کلونی مورچگان، پیشنهاد شده به وسیله‌ی پورزاهدی و ابوالقاسمی [۲]، به عنوان یکی از نمونه‌های پرارجاع در ادبیات، مبنا قرار می‌گیرد. این الگوریتم به وسیله‌ی الگوی ارباب-کارگر موازی‌سازی شده و نتایج مربوطه گزارش و تحلیل می‌شوند. این بخش، ابتدا الگوریتم موازی مورد پیشنهاد این مطالعه را برای حل مساله طراحی شبکه گسسته حمل‌ونقل تشریح نموده، و سپس به بیان علت انتخاب الگوی ارباب-کارگر در حل موازی این مساله می‌پردازد.

٤-۱- الگوریتم موازی پیشنهادی

الگوی موازی‌سازی ارباب-کارگر پیشنهادی در این مطالعه، جزییات یک الگوریتم فراابتکاری مشابه با شکل (۱) را دارد، با این تفاوت که بیش از یک پردازنده دراختیار بوده، که هر پردازنده در نقش یک عامل مورچه در الگوریتم کلونی مورچگان عمل می‌کند. در این طراحی، اطلاعات اصلی الگوریتم کلونی مورچگان، از جمله میزان فرمون-گذاری و بهترین جواب جاری، در اختیار پردازنده ارباب قرار داشته، این پردازنده وظیفه بهنگام‌سازی این اطلاعات را برعهده‌دارد.

به منظور بهنگام‌سازی اطلاعات الگوریتم، پردازنده ارباب از طریق ایجاد جواب‌های امکان‌پذیر جدید و ارسال آن‌ها به پردازنده‌های کارگر، آنان را در راستای انجام محاسبات ارزیابی جواب‌های ایجاد شده به خدمت می-گیرد. این فرآیند را یک تکرار موازی می‌نامیم (که شامل حلقه داخلی فلوچارت شکل (۱) است). هر پردازنده کارگر، تنها یک وظیفه دارد و آن، این‌است که در هر تکرار موازی یک جواب امکان‌پذیر از ارباب دریافت نموده، آن را ارزیابی کند و حاصل ارزیابی خود را به پردازنده ارباب برگرداند.

الگوریتم موازی پیشنهادی را می‌توان با بیان دیگری برمبنای فلوچارت شکل (۱) نیز تشریح نمود. با نگاهی به این فلوچارت، می‌توان گفت در الگوریتم پیشنهادی پردازنده ارباب تمامی مراحل، به استثنای مرحله ارزیابی جواب‌ها، را بر عهده دارد. تنها در مرحله ارزیابی است که پردازنده ارباب وظیفه ارزیابی k امین جواب ایجاد شده را به پردازنده کارگر k ام (در نقش مورچه k ام) واگذار می-کند. لازم به ذکر است که در این مطالعه، پردازنده‌های کارگر به منظور ارزیابی جواب‌ها از الگوریتم تکراری ترکیب کوژ با ۲۰ تکرار برای تخصیص ترافیک استفاده می‌کنند.

٤-۲- انتخاب الگوی موازی سازی ارباب-کارگر

دستیابی به یک الگوریتم موازی موثر، از نوع ارباب کارگر، نیازمند کاهش هر چه بیشتر سربار ارتباطات و سربار بیکاری بین پردازنده‌هاست. به منظور کاهش سربار ارتباطات، در مطالعه حاضر، اطلاعات حجیم مورد نیاز پردازنده‌های کارگر، شامل شبکه حمل‌ونقل، پروژه‌ها، و همچنین تقاضای سفر بین گره‌های شبکه، به طور اولیه

برای همه آنها ذخیره می‌گردد. طی اجرای الگوریتم موازی، هر پردازنده کارگر یک جواب در قالب یک رشته دودویی از پردازنده ارباب دریافت نموده، شبکه در دست خود را مطابق این جواب بهنگام ساخته، مساله تخصیص ترافیک را بر روی این شبکه حل می‌کند، و در آخر یک عدد را به عنوان نتیجه ارزیابی خود به پردازنده ارباب برمی‌گرداند. به این ترتیب، می‌توان گفت که زمان ارتباط بین پردازنده‌ها برابر است با تبادل یک رشته دودویی و یک عدد، که این حجم عملیات در مقایسه با حجم عملیات ارزیابی (حل مساله تخصیص ترافیک) که به شکل موازی به وسیله‌ی کارگران انجام می‌شود، ناچیز است.

همچنین، ازآنجاکه پردازنده‌های کارگر در تکرارهای موازی، محاسبات مشابهی را (برابر با حل یک مساله تخصیص ترافیک) انجام می‌دهند، زمان بیکاری این پردازنده‌ها و در نتیجه، سربار بیکاری نیز ناچیز خواهد بود.

۵- نتایج عددی

این بخش، رفتار الگوریتم موازی پیشنهادشده را در دو گام بررسی می‌کند. در گام نخست، الگوریتم موازی کلونی مورچگان، با ۸ هسته پردازشی، در کنار همین الگوریتم در حالت تک‌هسته‌ای (الگوریتم پورزاهدی و ابوالقاسمی) بررسی و مقایسه می‌شود. در گام بعدی، رفتار الگوریتم موازی کلونی مورچگان، به عنوان یک الگوریتم موازی فراابتکاری، در کنار رفتار الگوریتم موازی شاخه‌وکرانه، به عنوان یک الگوریتم موازی دقیق، گزارش خواهد شد. این گام، علاوه بر فراهم ساختن بینشی از رفتار موازی الگوریتم‌های مورد نظر، می‌تواند با به دست آوردن جواب دقیق مساله راه را برای درک فاصله جواب‌های به‌دست‌آمده تا جواب دقیق مساله (بهینه جهانی) هموار سازد.

شبکه حمل‌ونقلی که این مطالعه برای اجرای برنامه‌ها درنظر می‌گیرد، شبکه شیکاگو خلاصه‌شده است. این شبکه، که از این پس از آن به اختصار به صورت شبکه شیکاگو یاد می‌شود، شبکه‌ای بزرگ و دارای ۹۳۳ گره است که از این تعداد ۳۸۷ گره تولیدکننده سفر است. شبکه شیکاگو در مجموع دارای ۲۹۵۰ یال و ۹۳۵۱۳ تقاضای سفر غیر صفر بین گره‌هاست. اطلاعات این شبکه همگی بر مبنای گزارش‌های مرجع [۲۶] مورد استفاده قرار گرفته‌اند.

پیشنهاد پروژه در شبکه شیکاگو، از طریق درنظر گرفتن ۱۶ بزرگراه دوطرفه، به طور فرضی، انجام گرفته‌است (درنظرگرفتن پروژه‌های فرضی، روشی‌است که پیش‌تر نیز در پژوهش‌های طراحی شبکه حمل‌ونقل، همچون مرجع [۲]، مورد استفاده قرار گرفته‌است). جدول (۱) جزییات پروژه‌های پیشنهادی بزرگراهی را، با فرض مشخصات یکسان در هر دو جهت، نمایش می‌دهد.

پروژه‌های پیشنهادی جدول (۱) بر اساس فاصله واقعی گره‌ها در شبکه شیکاگو، و به تقلید از یال‌های موجود این شبکه، با فرض بیشینه سرعت در بازه ۶۰ الی ۷۰ کیلومتر بر ساعت پیشنهاد شده‌اند (با توجه به دقت نه‌چندان بالای الگوریتم ترکیب کوژ در ملاحظه تغییرات جزیی در شبکه، تعریف این پروژه‌ها به‌گونه‌ای انجام‌گرفته که تاثیر اضافه-کردن هر یک از آن‌ها به تنهایی با الگوریتم ترکیب کوژ قابل‌مشاهده باشد). مقدار بودجه دردست برای ساخت پروژه‌ها برابر با ۱۲۰ واحد فرض شده‌است، که با توجه به مقادیر هزینه فرضی ساخت پروژه های پیشنهادی، یک سطح بودجه متوسط تلقی می‌شود.

پیش از گزارش مشاهدات و نتایج عددی، لازم به ذکر است که کلیه برنامه‌های مورد استفاده در این مطالعه، به زبان برنامه‌نویسی Java پیاده‌سازی و اجرا شده است. همچنین نتایج، همگی از اجرای برنامه‌ها روی یک خوشه ۸ هسته‌ای از پردازنده‌ها، از نوع Intel(R) Xeon(R) E5404 با

فرکانس کلاک ۲/۰۰GHz، و با دراختیارداشتن مقدار GB ۲٤/٦ حافظه، گزارش می‌شوند.

جدول (۱) پروژه‌های پیشنهادی ساخت در شبکه شیکاگو

شماره پروژه	بزرگراه دوطرفه بین گره‌ها		زمان سفر جریان آزاد (دقیقه)	ظرفیت عملی (وسیله بر ساعت)	هزینه (واحد بودجه)
۱	۱٤	۷	۲/٤۰	۹۰۰۰	۱٤/۱
۲	۱٤	۱۰	۲/۸۸	۱۰۰۰۰	۱٤/۹
۳	۱۰	۱۱	۳/۲٤	۹۵۰۰	۱۵/۱
٤	۱۳	۱٦	۲/۸۸	۱۰۵۰۰	۱۵/۲
۵	۱٤	۱٦	٤/۸۰	۸۵۰۰	۱۵/۹
٦	۱۲	۱۷	۵/۰٤	۸۵۰۰	۱٦/۱
۷	۵٦۸	۵٦۲	۲/۸۸	۱۱۵۰۰	۱۷/۳
۸	۱٤	۱۷	۵/۲۰	۱۰۰۰۰	۱۷/۳
۹	۳۵٦	۳۵۷	۷/٤۲	۹۰۰۰	۱۸/۵
۱۰	۳۸۷	۵۳٤	٦/٦۰	۱۰۵۰۰	۱۹/۱
۱۱	٤۹٦	٤۹۰	۷/۰۲	۱۱۰۰۰	۲۰/۱
۱۲	٦۰۵	۳۹۹	۵/۳۷	۱۳۵۰۰	۲۰/۳
۱۳	۵۷۵	۵٦۸	۵/۹۹	۱۳۵۰۰	۲۱/۲
۱٤	۷٦٦	۷۷۹	۱۰/٦۱	۹۰۰۰	۲۱/۲
۱۵	۵۳۵	٤۳۵	۱۳/۰۹	۹۵۰۰	۲٤/۳
۱٦	۵۸۰	۵٦۸	۱۰/۱۰	۱٤۰۰۰	۲۸/۲

۵-۱- رفتار الگوریتم موازی کلونی مورچگان

الگوریتم‌های فراابتکاری در مدت اجراهای مختلف لزوما عملکرد یکسانی از خود نشان نمی‌دهند. ازاین‌رو، برنامه‌های موازی و تک‌هسته‌ای کلونی مورچگان، هریک در این مطالعه سه بار اجرا شدند. این تعداد اجرا، البته ناشی از محدودیت منابع دردست نویسندگان بوده‌است و انجام قضاوت‌های کلی و نتیجه‌گیری‌های دقیق‌تر تنها با اجراهای بیشتر، برروی شبکه‌های گوناگون میسر می‌شود. سقف زمان اجرا در حالت موازی ٤ هزار ثانیه و در حالت تک‌هسته‌ای ۱۰ هزار ثانیه درنظر گرفته‌شد. جدول‌های (۲) و (۳) رکوردی از جزییات چگونگی بهبود تابع هدف را طی زمان برای اجراها به ترتیب در حالت تک‌هسته‌ای و موازی نشان می‌دهند.

در مدت اجراهای یادشده، اطلاعاتی مانند زمان‌های بهبود تابع هدف و مشخصات جواب ذخیره شدند. بهترین جواب جاری در جدول‌های (۲) و (۳) به صورت یک رشته دودویی به طول ۱٦ نمایش داده‌شده‌است. ۱ یا ۰ بودن عدد k ام در این رشته‌ها به ترتیب به‌منزله تصمیم به ساخت یا عدم ساخت پروژه k ام در جواب مربوطه است (۱٦تا۱=k).

همچنین گزارش درصد خطای جواب‌های یافت‌شده تا جواب بهینه جهانی مساله، نیازمند به دراختیارداشتن جواب بهینه جهانی است. این اطلاع مهم، از اجرای الگوریتم موازی شاخه‌وکرانه حاصل شده‌است که در بخش ۵-۲ به آن اشاره خواهد شد.

چنان‌که در سطر اول گزارش‌های اجرا ملاحظه می‌شود، فاصله اولیه الگوریتم‌ها تا جواب بهینه (در وضعیت عدم ساخت همه پروژه‌ها) برابر با ۲/۵٤ درصد است. در نتیجه تلاش الگوریتم‌ها، معطوف به نزدیک‌ساختن هرچه بیشتر این فاصله به ۰ درصد می‌شود. براساس جدول‌های (۲) و (۳)، ملاحظه می‌شود که همه اجراهای تک‌هسته‌ای و موازی، به جز یک مورد (اجرای اول در حالت تک‌هسته‌ای)، توانسته‌اند در پایان به خطایی زیر ۰/۱ درصد تا جواب بهینه جهانی دست پیدا کنند (این مشاهده، خود حاکی از قدرت بالای الگوریتم فراابتکاری پورزاهدی و ابوالقاسمی [۲] در حل مساله روی شبکه‌های نسبتا بزرگ است).

گزارش اجراهای تک‌هسته‌ای در جدول (۲) نشان می‌دهد که این برنامه‌ها در یک مورد به جواب بهینه جهانی مساله دست‌یافته و در دو مورد دیگر به مقادیر خطایی معادل ۰/۰۷ و ۰/۱۷ درصد رسیده‌اند. اما اجراهای موازی، مطابق جدول (۳)، توانسته‌اند در دو مورد از سه اجرا، در نهایت به جواب بهینه جهانی مساله دست‌پیداکنند، و در یک مورد دیگر نیز، به فاصله ۰/۰۷ درصدی از این جواب

برسند. این مشاهده، با توجه به فرصت زمان اجرای کمتر داده‌شده به برنامه‌های موازی، نشان‌دهنده تاثیر آشکار موازی‌سازی در بهبود عملکرد الگوریتم است.

به منظور ارائه تصویر بهتری از چگونگی بهبود (کاهش) تابع هدف در طی زمان حل، شکل (۲) این بهبود را در طی زمان برای اجراهای تک‌هسته‌ای (شکل الف) و موازی (شکل ب)، و در کنار مقدار جواب بهینه جهانی نمایش می‌دهد. در شکل (۲) منحنی‌ها از اولین بهبود تابع هدف به بعد، در دومقیاس افقی زمان اجرا، یکی از ۰ تا ۱۰ هزار ثانیه و دیگری از ۰ تا ٤ هزارثانیه، ترسیم شده‌اند.

جدول (۲) اجرای تک‌هسته‌ای الگوریتم کلونی مورچگان

خطا تا جواب بهینه جهانی (٪)	هزینه (واحد بودجه)	بهترین جواب جاری	تابع هدف مساله (وسیله-دقیقه)	زمان اجرا (ثانیه)
اجرای شماره ۱				
۲/٥٤	۰/۰	[۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰]	۱۸۳۹۰۵۰۲	۰
۰/۱۹	۱۱۸/۷	[۰۱۱۰۱۱۱۰۰۱۰۱۰۰۰۰]	۱۷۹٦۸٤۳۱	۹۰
۰/۱۷	۱۱۷/۰	[۱۰۱۱۱۰۱۰۰۱۰۱۰۰۰۰]	۱۷۹٦٤٦۰۸	۱٥۱٦
اجرای شماره ۲				
۲/٥٤	۰/۰	[۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰]	۱۸۳۹۰۵۰۲	۰
۰/۳٦	۱۱۹/۹	[۰۱۰۰۰۰۱۰۰۱۱۱۰۰۰۱]	۱۷۹۹۸۹۰۳	۱۰۲
۰/۳٦	۱۱٥/۸	[۰۱۱۰۰۰۱۰۰۱۰۰۱۰۰۱]	۱۷۹۹۸۲۷٥	۲۹۹
۰/۳۰	۱۱۹/۹	[۰۰۱۰۱۰۱۰۰۱۰۰۰۰۱۱]	۱۷۹۸۷۷٦۷	٦٤۲
۰/۲۹	۱۲۰/۰	[۱۱۱۱۰۰۱۰۰۱۰۰۰۰۱۰]	۱۷۹۸٦۳۹۹	۸۱۰
۰/۲۷	۱۱۷/۸	[۰۱۰۰۱۰۱۰۰۰۰۱۰۱۰۱]	۱۷۹۸۳٤۲۳	۲۲۹۰
۰/۲۷	۱۱۸/۹	[۱۱۰۱۱۰۱۰۰۰۰۱۰۱۰۰]	۱۷۹۸۲٤۸٥	۳۱۸٤
۰/۲۰	۱۲۰/۰	[۱۰۱۱۱۰۱۰۰۰۰۰۱۱۰۰]	۱۷۹٦۹۷٤۱	۳٦٥۰
۰/۱۸	۱۱٥/۷	[۰۱۰۰۱۰۱۰۰۱۰۱۰۰۰۱]	۱۷۹٦۷٤۸٥	٥۱۷۲
۰/۰۰	۱۱۸/۷	[۰۱۱۱۱۰۱۰۰۱۰۰۱۰۰۰]	۱۷۹۳٤۳۹٦	۸۳۰۹
اجرای شماره ۳				
۲/٥٤	۰/۰	[۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰]	۱۸۹۳۰٥۰۲	۰
۰/٥۰	۱۰۷/۳	[۰۱۱۰۰۰۱۰۱۰۰۱۱۰۰۰]	۱۸۰۲۳۳۱۲	۹۰
۰/۳۹	۱۱٤/۰	[۰۱۰۰۰۰۱۰۰۱۰۱۱۱۰۰]	۱۸۰۰۳۹٥۳	۱۷٤
۰/۳۸	۱۱٦/۹	[۰۰۰۱۱۰۱۰۰۱۰۰۱۰۰۱]	۱۸۰۰۲۷۷۹	٥۹۷
۰/۲۷	۱۱٦/۰	[۰۰۰۱۱۰۱۰۰۱۰۱۰۰۰۱]	۱۷۹۸۳٤۰۳	۷۸۷
۰/۱٦	۱۱۹/٦	[۰۱۱۰۱۱۱۰۰۱۰۰۱۰۰۰]	۱۷۹٦۳٥٤۷	۲۰۱۷
۰/۰۷	۱۱۹/۹	[۰۱۱۱۱۰۱۰۰۰۰۱۱۰۰۰]	۱۷۹٤٦۷۲۱	٦٦۳۲

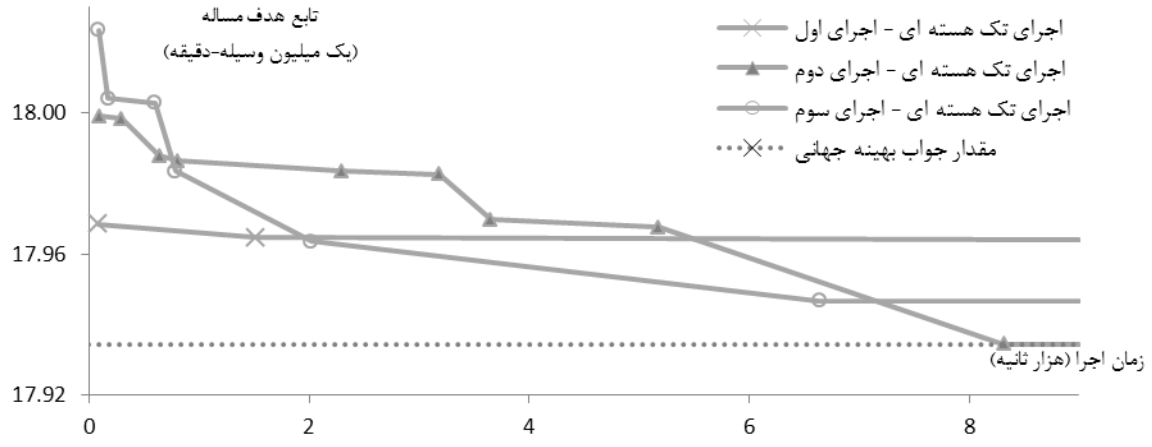
جدول (۳) اجرای موازی الگوریتم کلونی مورچگان

خطا تا جواب بهینه جهانی (٪)	هزینه (واحد بودجه)	بهترین جواب جاری	تابع هدف مساله (وسیله-دقیقه)	زمان اجرا (ثانیه)
اجرای شماره ۱				
۲/٥٤	۰/۰	[۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰]	۱۸۳۹۰٥۰۲	۰
۰/۳٥	۱۱٦/٦	[۰۱۰۰۱۰۱۰۰۱۰۰۱۰۰۱]	۱۷۹۹۷٥٤۸	۲۰
۰/۱۸	۱۱۹/۷	[۰۱۰۱۱۱۱۰۰۱۰۰۱۰۰۰]	۱۷۹٦٦۷٥۷	٤٦
۰/۱٦	۱۱۹/٦	[۰۱۱۰۱۱۱۰۰۱۰۰۱۰۰۰]	۱۷۹٦۳٥٤۷	۲٥۱
۰/۰۷	۱۱۹/۹	[۰۱۱۱۱۰۱۰۰۰۰۱۱۰۰۰]	۱۷۹٤٦۷۲۱	۲۷۷
۰/۰۱	۱۱۷/۸	[۰۱۱۱۱۰۱۰۰۱۰۱۰۰۰۰]	۱۷۹۳٥۳۸۱	٥۳۷
۰/۰۰	۱۱۸/۷	[۰۱۱۱۱۰۱۰۰۱۰۰۱۰۰۰]	۱۷۹۳٤۳۹٦	۳۷٦۳
اجرای شماره ۲				
۲/٥٤	۰/۰	[۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰]	۱۸۳۹۰٥۰۲	۰
۰/۲۰	۱۲۰/۰	[۱۰۱۱۱۰۱۰۰۰۰۰۱۱۰۰]	۱۷۹٦۹۷٤۱	۲۲
۰/۱۲	۱۱۹/۹	[۰۱۱۱۱۰۱۰۰۰۰۱۰۱۰۰]	۱۷۹٥٥۸۹۰	٦٥
۰/۰۷	۱۱۹/۹	[۰۱۱۱۱۰۱۰۰۰۰۱۱۰۰۰]	۱۷۹٤٦۷۲۱	۲٤۸۷
اجرای شماره ۳				
۲/٥٤	۰/۰	[۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰]	۱۸۳۹۰٥۰۲	۰
۰/۳٦	۱۱٥/٤	[۱۱۱۱۰۰۱۰۱۰۰۱۰۰۰۰]	۱۷۹۹۹٥۱۹	۲۰
۰/۳۳	۱۱۲/٦	[۰۱۱۰۱۰۱۰۰۰۰۰۱۰۰۱]	۱۷۹۹۳۰۰۳	٤٥
۰/۱۱	۱۱۷/٦	[۱۱۱۰۱۰۱۰۰۱۰۰۱۰۰۰]	۱۷۹٥٤۳٦۱	۷۱
۰/۰۷	۱۱۹/۹	[۰۱۱۱۱۰۱۰۰۰۰۱۱۰۰۰]	۱۷۹٤٦۷۲۱	۳۸۲
۰/۰۰	۱۱۸/۷	[۰۱۱۱۱۰۱۰۰۱۰۰۱۰۰۰]	۱۷۹۳٤۳۹٦	۱۳۸۳

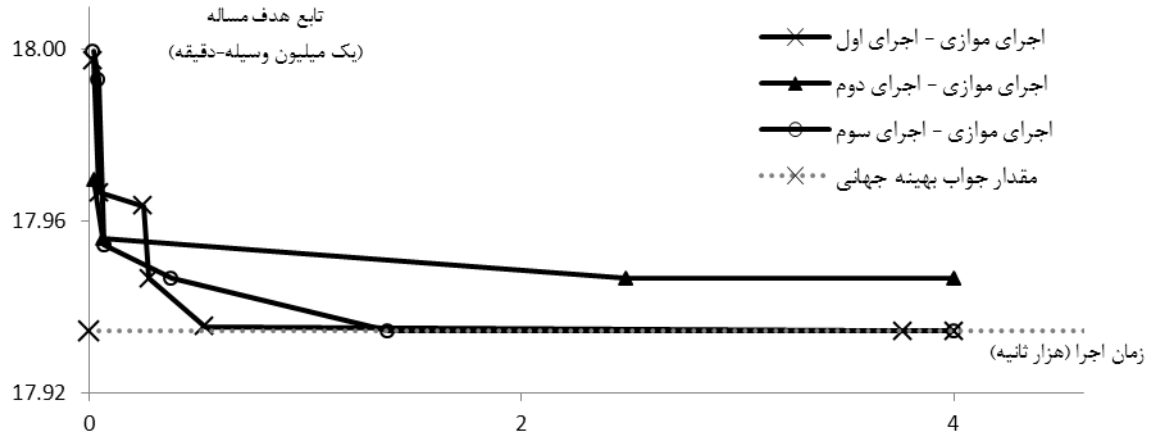
چنان‌که در این شکل ملاحظه می‌شود، الگوی کلی کاهش تابع هدف در طی زمان اجرا مشابه به نظر می‌رسد. اما این کاهش در حالت موازی، در مقایسه با حالت تک‌هسته‌ای، به‌طرز محسوسی با زمان اجرای کمتری انجام می‌پذیرد. این نکته از تفاوت مقیاس افقی منحنی‌های مربوط به اجراهای موازی در مقایسه با اجراهای تک‌هسته‌ای در شکل (۲) قابل مشاهده است.

۵-۲- مشاهده رفتار در کنار رفتار الگوریتم موازی شاخه‌وکرانه

با اطلاع از جواب بهینه جهانی مساله، شرایط برای بررسی میزان خطای الگوریتم تا بهترین جواب امکان‌پذیر فراهم شد (چنان‌که در بخش ۵-۱ ملاحظه شد).



الف: اجراهای تک‌هسته‌ای



ب: اجراهای موازی

شکل (۲) بهبود تابع هدف در الگوریتم‌های تک‌هسته‌ای و موازی کلونی مورچگان

این اطلاع در این مقاله از طریق اجرای یک الگوریتم موازی شاخه‌وکرانه [۲۱] حاصل گردید. اجرای الگوریتم-های موازی دقیق، علاوه بر کمک به تحلیل‌های یادشده، می‌تواند در مشاهده رفتار موازی الگوریتم‌های فراابتکاری، در کنار یک الگوریتم حل دقیق، مفید واقع شود. البته باید توجه داشت که هدف این بخش انجام مقایسه بین رفتار دو الگوریتم نیست، زیرا، این دو الگوریتم از دو خانواده متفاوت (یکی فراابتکاری و دیگری دقیق) بوده و به‌طور رایج با یکدیگر مورد مقایسه قرار نمی‌گیرند.

همچنین، درصورت نیاز به مقایسه الگوریتم موازی کلونی مورچگان با الگوریتم دقیق شاخه‌وکرانه، باید رفتار غیرقطعی الگوریتم فراابتکاری درنظر گرفته‌شده، این الگوریتم به تعدادی بسیار بیشتر از سه مورد اجرا شده و رفتار متوسط آن درنظرگرفته‌شود. در نتیجه، این بخش فقط رفتار دو الگوریتم موازی مورچگان و شاخه‌وکرانه را در کنار همدیگر گزارش نموده، از مقایسه و قضاوت کلی در این خصوص پرهیز می‌نماید.

برای این منظور، کافی‌است مانند الگوریتم‌های فراابتکاری در بخش ۵-۱، در مطالعه زرین‌مهر، رکوردی از چگونگی بهبود تابع هدف الگوریتم موازی شاخه‌وکرانه در طی زمان

اجرا ذخیره گردد. جدول (٤) این رکورد را برای الگوریتم موازی شاخه‌وکرانه گزارش می‌کند. لازم به ذکر است که این اجرا تنها یک بار صورت‌گرفته‌است، زیرا این الگوریتم در اجراهای مختلف عملکردهای متفاوتی نشان نمی‌دهد [٢٤].

مطابق انتظار، در جدول (٤) مشاهده می‌شود که الگوریتم موازی شاخه‌وکرانه در نهایت با خطای ۰ درصد (یعنی به طور دقیق) به جواب مساله دست پیدا می‌کند. این جواب در قالب یک رشته دودویی به صورت [۰۱۱۱۱۰۱۰۰۱۰۰۱۰۰۰] و دارای مقدار تابع هدف بهینه ۱۷۹۳٤۳۹٦ وسیله-دقیقه در شبکه است. اگرچه این جواب بهینه در زمان ۱۳۷۲۳ ثانیه برای اولین بار به وسیله‌ی الگوریتم موازی شاخه‌وکرانه ملاقات می‌شود، اما الگوریتم برای حصول اطمینان از بهینگی جواب، نیازمند به ادامه جستجو تا مدت زمان ۳۲٤۳٦ ثانیه است. شکل (۳) تصویری از رفتار موازی الگوریتم شاخه‌وکرانه در کنار سه مورد اجرای موازی الگوریتم کلونی مورچگان ارائه می‌دهد.

به منظور درک بهتر تفاوت عملکرد دو الگوریتم، قسمت (ب) از شکل (۳) رفتار دو الگوریتم را در مقیاس مدت‌زمان اجرای ۱۵۰۰ ثانیه نمایش می‌دهد. در این شکل نیز می‌توان مشاهده کرد در حالی که تابع هدف الگوریتم موازی شاخه وکرانه به کندی در جهت بهینگی در حال کاهش است، الگوریتم کلونی مورچگان به سرعت به جواب نهایی خود همگرا می‌شود. البته مجددا تاکید می‌شود که مقایسه کلی در این خصوص، به اجراهای بیشتر الگوریتم موازی کلونی مورچگان نیازمند است.

٦- نتیجه‌گیری

مقاله حاضر، به حل مساله طراحی شبکه گسسته حمل-ونقل، از طریق الگوریتم موازی کلونی مورچگان پرداخت. نتایج این مقاله را می‌توان به این ترتیب خلاصه نمود:

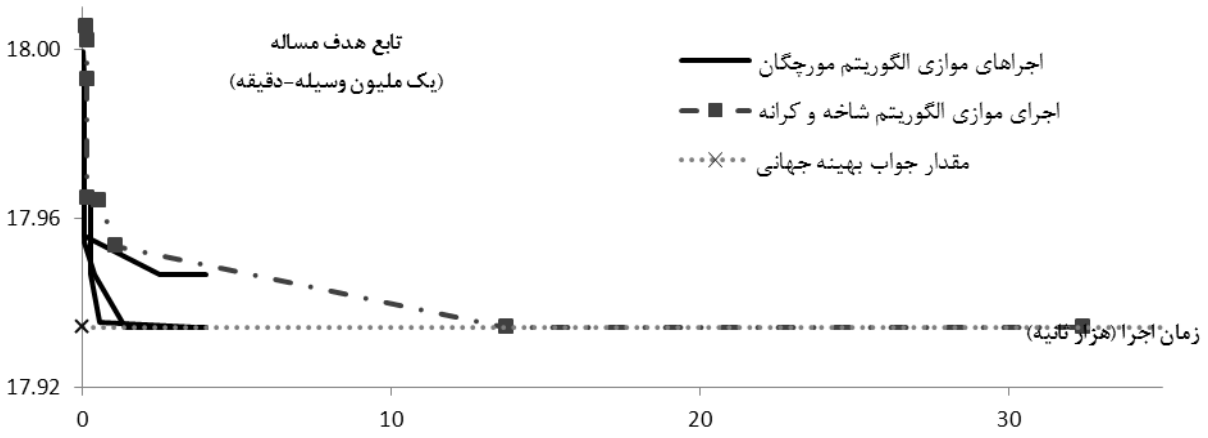
۱- الگوریتم فراابتکاری کلونی مورچگان، به دلیل ناچیز بودن سربارهای ارتباطات و بیکاری بین پردازنده‌ها، قابل موازی‌سازی با یک الگوی ارباب-کارگر است. در الگوی پیشنهادی این مقاله، یک پردازنده (ارباب) اطلاعات اصلی الگوریتم را در اختیار داشته و در هر تکرار، سایر پردازنده-ها (کارگران) را در راستای ارزیابی‌کردن جواب‌های ایجاد-شده به‌خدمت‌می‌گیرد. این الگوریتم در حالت موازی با ۸ هسته‌پردازشی پیاده‌سازی و عملکرد آن در شبکه شیکاگو خلاصه‌شده با ١٦ پروژه بررسی و گزارش گردید.

جدول (٤) اجرای موازی الگوریتم شاخه و کرانه

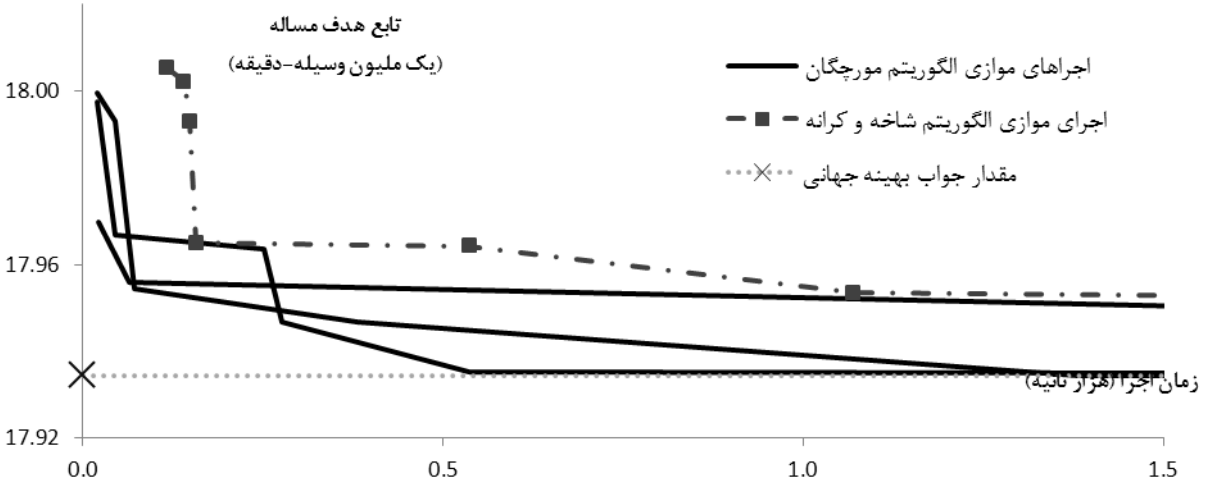
خطا تا جواب بهینه جهانی (٪)	هزینه (واحد بودجه)	بهترین جواب جاری	تابع هدف مساله (وسیله-دقیقه)	زمان اجرا (ثانیه)
۲/٥٤	۰/۰	[۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰]	۱۸۳۹۰٥۰۲	۰
۰/٤۰	۱۰۸/٦	[۱۱۱۱۱۱۱۰۰۰۰۰۰۰۰]	۱۸۰۰٥۲۹۰	۱۱۸
۰/۳۸	۱۰۹/۸	[۱۱۱۱۱۰۱۱۰۰۰۰۰۰۰۰]	۱۸۰۰۲۰۸۲	۱٤۱
۰/۳۳	۱۱۱/۰	[۱۱۱۱۱۰۱۰۱۰۰۰۰۰۰۰]	۱۷۹۹۲۹٤٥	۱٤۹
۰/۱۷	۱۱۱/٦	[۱۱۱۱۱۰۱۰۰۱۰۰۰۰۰۰]	۱۷۹٦٤۹٤۰	۱٥۸
۰/۱۷	۱۱٦/۰	[۱۱۱۱۰۰۱۰۰۱۰۱۰۰۰۰]	۱۷۹٦٤۲۸۷	٥۳۷
۰/۱۱	۱۱٦/۷	[۱۱۱۰۱۰۱۰۰۱۰۱۰۰۰۰]	۱۷۹٥۳٤۸۱	۱۰۷۰
۰/۰۰	۱۱۸/۷	[۰۱۱۱۰۰۱۰۰۱۰۰۱۰۰۰]	۱۷۹۳٤۳۹٦	۱۳۷۲۰
۰/۰۰	۱۱۸/۷	[۰۱۱۱۱۰۱۰۰۱۰۰۱۰۰۰]	۱۷۹۳٤۳۹٦	۳۲٤۳٦

۲- در کنار الگوریتم موازی کلونی مورچگان، یک الگوریتم موازی شاخه‌وکرانه موجود در ادبیات نیز به عنوان یک الگوریتم موازی دقیق به اجرا گذاشته‌شد. این اجرا موجب امکان اعتبارسنجی برای الگوریتم فراابتکاری مورچگان شد.

۳- الگوریتم موازی کلونی مورچگان، سه بار، با دادن فرصت ٤ هزار ثانیه‌ای به اجرا درآمد. از این سه مورد، در دو مورد این الگوریتم موفق به یافتن جواب دقیق مساله شد و در یک مورد به جوابی با فاصله ۰/۰۷ از جواب دقیق مساله دست‌یافت.



(الف) اجرا تا زمان اتمام کل برنامه‌ها



(ب) اجرا تا زمان ۱۵۰۰ ثانیه

شکل (۳) بهبود تابع هدف در الگوریتم موازی شاخه‌وکرانه و سه اجرای موازی الگوریتم کلونی مورچگان

درحالی‌که الگوریتم تک‌هسته‌ای کلونی مورچگان، با دادن فرصتی بیشتر، به مدت ۱۰ هزار ثانیه، از سه اجرا، در یک اجرا جواب دقیق مساله را پیدا کرده و در دو اجرای دیگر، در فاصله‌های ۰/۰۷ و ۰/۱۷ درصدی از جواب دقیق متوقف شد.

٤- الگوریتم موازی شاخه‌وکرانه، با صرف بیش از ۳۲ هزار ثانیه به جواب دقیق مساله دست‌یافت و رفتار موازی این الگوریتم دقیق، درکنار الگوریتم موازی کلونی مورچگان گزارش شد.

۷- پیشنهادها

۱- مقایسه و قضاوت کلی در خصوص رفتار موازی الگوریتم کلونی مورچگان، با توجه به رفتار غیرقطعی آن، نیازمند به اجراهای متعدد این الگوریتم است. این مطالعه، با توجه به محدودیت در دسترسی به منابع، تنها موفق به انجام سه اجرای تک‌هسته‌ای و موازی شد. انجام اجراهای بیشتر، بر روی شبکه‌های حمل‌ونقلی گوناگون، می‌تواند راه را برای قضاوت‌های کلی‌تر هموار سازد.

۲- با توجه به رواج کاربرد سیستم‌های موازی‌سازی

قدرتمند با هسته‌های پردازشی فراوان، کاربرد سایر الگوهای موازی‌سازی و حل موازی مسائل بزرگ طراحی شبکه حمل‌ونقل در سطح این سیستم‌ها جالب‌توجه است. همچنین، درایورت، سایر الگوریتم‌های قدرتمند تخصیص ترافیک، می‌توانند مورد استفاده قرار گیرد.

۸- قدردانی

نویسندگان، این مقاله را به روح بلند دوست ازدست‌داده خود، مهندس فرزاد فرخزاده، فارغ‌التحصیل دانشکده مهندسی برق-کامپیوتر از دانشکده‌فنی دانشگاه‌تهران، تقدیم می‌کنند.

۹- مراجع

[1] Alba, E. *Parallel metaheuristics: A new class of algorithms*, Wiley Series on Parallel and Distributed Computing, 2005.

[2] Poorzahedy, H., Abulghasemi, F. *Application of ant system to network design problem*, Transportation, 32(3), 2005, 251-273.

[3] Garey, M. R., Johnson, D. S. *Computers and intractability: a guide to the theory of NP-completeness*, W. H. Freeman and Co, New York, 1979.

[4] Ben-Ayed, O., Boyce, D. E., Blair III, C. E. *A General Bilevel Linear Programming Formulation of the Network Design Problem*, Transportation Research Part B, 22(4), 1988, 311-318.

[5] Vitins, B. J., Axhausen, K. W. *Optimization of large transport networks using the ant colony heuristic*, Computer-Aided Civil and Infrastructural Engineering, 24(1), 2009, 1-14.

[6] Pradhan, A., Mahinthakumar, G. *Finding all-pairs shortest path for a large-scale transportation network using parallel Floyd-Warshal and parallel Dijkstra algorithms*, Journal of Computing in Civel Engineering, 27(3), 2013, 263-273.

[7] Yang, H., Bell, M. G. H. *Models and algorithms for road network design: a review and some new developments*, Transport Reviews-Transnational Transdisciplinary Journal, 18(3), 1998, 257-278.

[8] Babazadeh, A., Poorzahedy, H., Nikoosokhan, S. *Application of particle swarm optimization to transportation network design problem*, Journal of King Saud University-Science, 23(3), 2011, 293-300.

[9] Sheffi Y. *Urban transportation networks: equilibrium analysis with mathematical programming methods*, Prentice Hall, 1985.

[10] Aashtiani, H. Z. *The multi-modal traffic assignment problem*, Ph.D. Dissertation, MIT, 1979.

[11] Nie, Y. *A class of bush-based algorithms for the traffic assignment algorithm*, Transportation Research Part B, 44(1), 2010, 73-89.

[12] Izadpanah, P., Aashtiani, H. Z. *Application of Paths Information in Network Design Problem*, Journal of Transportation Engineering, 138(7), 2011, 863-870.

[13] LeBlanc, L. J., Morlok, E.K., Pierskalla, W.P., *An efficient approach to solving the road network equilibrium traffic assignment problem*, Transportation Resesarch, 9(5), 1975, 309-318.

[14] Vitins, B. J., Axhausen, K. W. *Patterns and grammars for transport network generation*, Proceedings of 14th Swiss Transportation Research Conference, 2010.

[15] Farahani, R. Z., Miandoabchi, E., Szeto, W. Y., Rashidi, H. *A review of urban transportation network design problems*, European Journal of Operational Research, 229, 2013, 281-302.

[16] Yin, Y. *Genetic-algorithms-based approach for bi-level programming models*. Journal of Transportation Engineering, 126(2), 2000, 115-120.

[17] Lee, C. K., Yang, K. I. *Network design on one-way streets with simulated annealing*, Papers in Regional Science, 73(2), 1994, 119–134.

[۱۸] ابوالقاسمی؛ فرهاد؛ *کاربرد الگوریتم سیستم مورچه‌ها در مساله طراحی شبکه*؛ پایان‌نامه کارشناسی‌ارشد، موسسه عالی پژوهش در برنامه‌ریزی توسعه؛ ۱۳۸۰.

[19] Barney, B. *Introduction to parallel computing*, Lawrence Livermore National Laboratory, 2010.

[20] Gendron, B., Crainic, T.G. *Parallel branch-and-bound algorithms: survey and synthesis*, Operations Research, 42(6), 1994, 1042-1066.

[21] Roucairol, Catherine. *Parallel processing for difficult combinatorial optimization problems*, European Journal of Operational Research, 92(3), 1996, 573-590.

[22] Agrawal, J., Mathew, T. V. *Transit route network design using parallel genetic algorithm*, Journal of Computing in Civil Engineering, 18(3), 2004, 248-256.

[23] Yang, Z., Yu, B., Cheng, C. *A parallel ant colony algorithm for bus network optimization*, Computer-Aided Civil and Infrastructure Engineering, 22(1), 2007, 44-55.

[۲٤] زرین‌مهر؛ امیرعلی؛ *کاربرد الگوریتم‌های شاخه‌وکرانه موازی در حل مساله طراحی شبکه گسسته*؛ پایان‌نامه کارشناسی‌ارشد، دانشگاه صنعتی‌شریف؛ ۱۳۹۰.

[25] LeBlanc, L. J. *An algorithm for the discrete network design problem*, Transportation Science, 9(3), 1975, 183-199.

[26] Bar-Gera, H. *Transportation network test problems*, http://www.bgu.ac.il/~bargera/tntp. Accessed in Septemper 2011.

Parallelization of Ant Colony Algorithm in Transportation Discrete Network Design

A. Zarrinmehr[1*], M. Parvizi Omran[2], Y. Shafahi[3], S. Seyedabrishami[4]

1- M.Sc. of Transportation Planning, Dept. of Civil and Environmental Eng., Sharif University of Technology
2- M.Sc. of Computer Engineering – Algorithms and Computation, University of Tehran
3- Prof., Dept. of Civil and Environmental Eng., Sharif University of Technology
4- Assistant Prof., Dept. of Civil and Environmental Eng., Tarbiat Modares University

amirali.zarrinmehr@modares.ac.ir

Abstract:
Transportation Discrete Network Design Problem (TDNDP) is defined as the problem of selecting a subset of proposed projects (i.e. new highways) for construction, while holding the budget constraint, so as to minimize the total travel time of the network users. TDNDP has been often known as a problem with the bi-level programming formulation. At the upper level, this formulation allows for finding the optimal selection of the projects, while taking into account the route choice behavior of network users at its lower level. Such a formulation falls into the category of problems in the NP-Hard complexity class. These are resource-intensive problems which have not been exactly solved yet with any efficient algorithms. As a result, the main body of TDNDP related literature has ignored the exact solution of the problem and addressed TDNDP through heuristic and meta-heuristic approaches. These approaches contribute to find a rather high quality solution in a reasonable amount of time.

Using heuristic and meta-heuristic algorithms is one way to overcome the complexity of NP-Hard problems like TDNDP. However, application of parallel computing still remains as another way to speed up the performance of such algorithms. Parallel computation aims at harnessing multiple computing resources, e.g. computer processors, to solve a certain problem. Different parallelization paradigms have been developed so far to parallelize solution algorithms. These paradigms generally address the two fundamental questions of how and when the required information should be exchanged among the processors. A master-slave (MS) parallelization paradigm is one of the basic paradigms in which one processor, namely the master, holds the main information of the problem. The master generates new jobs whenever needed, distributes them among other processors (i.e. slaves), and exploits them to work on the sent jobs. This paper is going to explore the application of parallel computation in a meta-heuristic algorithm in TDNDP. A parallel Ant Colony Algorithm (ACA), based on the study of Poorzahedy and Abulghasemi, is proposed with the MS parallelization paradigm. The Chicago Sketch transportation network is considered as a case study with 16 bi-directional proposed projects. The results are reported in three runs over a cluster of 8 processing cores for both single-core and parallel ACAs. According to the performances observed in this paper, parallel algorithms can achieve high quality solutions in 4000 seconds, while this happens for the single-core algorithms in 10000 seconds. The parallel ACA finds the exact solution of the problem in two instances out of three runs and in the other instance it converges to a solution with 0.07 percent error from the exact solution. The parallel performance of ACA is also reported along with that of the branch and bound algorithm. It is observed that the parallel branch and bound algorithm requires more that 32000 seconds running time to find the exact solution of the problem. More accurate comparisons, however, can only be achieved by running the single-core and parallel ACAs more than the three times used in this paper.

Keywords: Transportation Discrete Network Design, Ant Colony Algorithm, Parallel Computing, Master-Slave Paradigm.