


An evolutionary algorithm for broadcast scheduling in wireless multi hop networks

الگوریتم تکاملی برای زمانبندی همه پخش در شبکه های بی سیم چند گامی

چکیده

یک چالش فنی در گسترش موفقیت آمیز و بهره برداری از شبکه های بی سیم چندگامی (WMN)، استفاده موثر از پهنای باند محدود کانال است. یک روش برای حل این چالش پخش زمانبندی استفاده از کانال با استفاده از روش راه دسترسی چندگانه تقسیم زمانی (TDMA) است. سه الگوریتم تکاملی به نام های الگوریتم های ژنتیکی (GA)، الگوریتم ژنتیک سیستم ایمنی (IGA) و الگوریتم memetic (MA) در این مقاله برای حل زمانبندی همه پخش بوسیله روش TDMA در شبکه های بی سیم چندگامی WMN مورد استفاده قرار می گیرد.

اصطلاح memetic از واژه meme برداشت شده است

(meme): یک عنصر فرهنگی یا رفتاری است که به وسیله عوامل غیر ژنتیکی از نسلی به نسل دیگر منتقل می شود 

هدف به حداقل رساندن طول دوره TDMA و به حداکثر رساندن انتقال گره با کاهش زمان محاسبه است.

در مقایسه با GA و IGA، MA به طور فعال بهبود راه حل ها را نشان گرفته است و به صراحت از تمام دانش موجود در باره مشکل بهره برداری می کند. نتایج شبیه سازی در موارد متعدد مشکل، تایید می کنند که MA به طور قابل توجهی چندین الگوریتم اکتشافی و تکاملی از طریق حل مشکل benchmark شناخته شده از نظر کیفیت راه حل انجام می دهد، و همچنین اثر MA در استفاده بهینه از پهنای باند کانال نشان می دهد.

واژگان

Wireless multihop networks: شبکه های بی سیم چند گامی

Broadcast scheduling: زمانبندی همه پخش

Genetic algorithm: الگوریتم ژنتیک

Immune genetic algorithm: الگوریتم ژنتیک ایمنی

Memetic algorithm: الگوریتم memetic

مقدمه

در شبکه های بی سیم ad hoc، شبکه های تک گامی و چند گامی به زیرساخت از پیش ساخته، مانند روترها در شبکه های سیمی یا نقاط دسترسی در شبکه های بی سیم مدیریت شده تکیه نمی کند. در عوض، هر گره از طریق فورواردها داده به گره های دیگر در مسیر یابی سهیم هستند، بنابراین تشخیص اینکه کدام گره داده را فورواردها می کند بطور پویا بر اساس اتصال شبکه انجام می شود. در شبکه تک گامی، هر یک از

ایستگاه سیار (MS) می تواند به طور مستقیم با همه MS های دیگر ارتباط برقرار کند. در شبکه های multihop بی سیم، یک یا چند گره میانی در طول مسیر بسته ها را از طریق لینک های بی سیم دریافت و فرورارد می کند. شبکه های multihop بی سیم پوشش شبکه را گسترش دادند، اتصال و انتقال روی چندین لینک کوتاه را اصلاح کردند، که به قدرت انتقال و انرژی کمتری نسبت به آنچه که روی لینک های طولانی مورد نیاز بود نیاز دارد. همچنین این ارتباطات قوی، گسترش سریع را فراهم می کند و در محیط های پویا به سرعت پاسخ می دهد.

شکل 1 یک شبکه multihop بی سیم ساده را نشان می دهد، هر گره یک ایستگاه سیار را نشان می دهد و یک خط، اتصال دو گره را نشان می دهد که دو MS (ایستگاه سیار) در یک محدوده ارتباطی هستند.

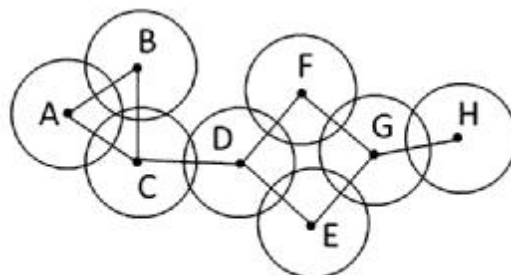


Fig. 1 A simple wireless multihop network

همسایه های A، MS هایی هستند که می توانند به طور مستقیم با A (به عنوان مثال، B و C) ارتباط برقرار کنند. تحرک گره در WMN باعث تغییرات مکرر در توپولوژی شبکه می شود. مشکل اصلی در طراحی WMN این است که همه MS ها نمی توانند به طور مستقیم با یکدیگر ارتباط برقرار کنند.

از اسلات های با طول زمانی ثابت تشکیل شده است که هر گره حداقل یک اسلات عمل انتقال را انجام می دهد. یک شبکه multihop بی سیم از تعدادی MS تشکیل شده است، که هر MS تعداد معینی همسایه MS دارد. فرض شده است که زمان به اسلات هایی تقسیم شده است، هر یک از مدت زمان ها برابر با حداکثر طول زمان انتقال بسته به علاوه حداکثر زمان انتشار بین دو MS تقسیم می شوند. فرض شده است که MS ها از آنتن های همه سویی یا omnidirectional استفاده می کنند. فرض شده است که کانال های بی سیم بدون نویز هستند و یک دریافت یا reception ناموفق تنها بدلیل برخورد یا collision است. MS ها در حالت نیمه دو طرفه یا half-duplex عمل می کنند، به عنوان مثال، یک MS در یک زمان می تواند ارسال کند و یا دریافت کند، اما نمی تواند هر دو کار را انجام دهد. دو یا چند MS بدون برخورد یا nonconflict می تواند در یک اسلات می توانند مشترک باشند. برخوردها در شبکه های multihop ممکن است به دو روش رخ دهد: برخوردهای اولیه و برخوردهای ثانویه

برخوردهای اولیه زمانی اتفاق می افتد که دو گره متصل به طور همزمان عمل انتقال را انجام دهند.

برخوردهای ثانویه زمانی اتفاق می افتد که دو یا چند بسته در یک بازه زمانی واحد به یک گره می رسند. این زمانی اتفاق خواهد افتاد که دو گره در فاصله دو پرش اجازه انجام عمل انتقال به طور همزمان را دارند. پس، گره میانی دو بسته متفاوت از دو گره ای که به طور مستقیم به آن متصل اند را در یک بازه زمانی دریافت خواهید کرد. دو MS، اگر در فاصله بیش از دو گره از هم قرار داشته باشند می توانند همان زمان بدون تداخل متقابل عمل انتقال را انجام دهند.

علاوه بر ویژگی های زمانبند ذکر شده در بالا، مطلوب است یک زمانبند WMN دارای ویژگی اطلاعات مورد نیاز اتصال کم باشد به عنوان مثال، برخی از الگوریتم ها به اطلاعات مربوط به اتصال به شبکه global نیاز دارند در حالی که برخی دیگر تنها به (به عنوان مثال، یک یا دو پرش) اطلاعات اتصال محلی یا local نیاز دارند. از آنجا که مخایره کردن این اطلاعات پهنای باند مصرف می کند، این اطلاعات باید حداقل باشد. مسائل اساسی محاسباتی و الگوریتمی در مشکل زمانبندی همه پخش شبکه های multihop بی سیم در فصل 16 مرجع [1] مورد بحث قرار گرفته است. اکثر الگوریتم های زمانبندی همه پخش با تولید یک زمانبندی اسمی طول محدود که در آن به هر ایستگاه حداقل یک اسلات برای انتقال اختصاص داده است و سپس به طور نامحدود این زمانبندی اسمی را تکرار می کند کار می کنند. این مسأله در NP-complete مرجع [2، 3] ثابت شده است. اکثر زمانبند های WMN فعال ساز گره و یا فعال ساز لینک [4] هستند، یک مثال از الگوریتم فعال سازی گره در [2] آمده است. یک زمانبند فعال ساز گره، گره هایی را برای انتقال انتخاب می کند که اطمینان دارد که تمام همسایگان آن، یک بسته را از هر گره به درستی دریافت خواهند کرد. یک زمانبند فعال ساز لینک گره هایی را برای انتقال انتخاب می کند که تضمین می کنند که گره مقصد با موفقیت بسته را دریافت می کند. الگوریتم های مختلف برای حل مشکل زمانبندی [2، 3، 5-17] پیشنهاد شده است. این الگوریتم ها طبقه بندی شده است به نظری گراف [2 و 18]، رنگ آمیزی گراف [19] و روش های احتمالاتی مانند گداختگی میدان متوسط [3]، و جستجو ممنوع [10]، الگوریتم ژنتیک [7، 9، 12]، الگوریتم شبکه های عصبی [11، 20] و با روش ژنتیکی عصبی ترکیب شده است [8].

بسیاری از این الگوریتم ها بر اساس هریک از دو نقطه هستند: یکی طول قاب را بدون در نظر گرفتن استفاده اسلات به حداقل می رساند و دیگری تلاش می کند استفاده از اسلات را در یک فرم به حداکثر برساند. بهینه سازی دو هدف به طور جداگانه منجر به یک راه حل خوب نیست.

رویکرد بهتر در نظر گرفتن هر دو این معیارها در روش یکپارچه برای حل مشکل زمانبندی همه پخش است. الگوریتم در [8] ترکیبی از یک شبکه عصبی هاپفیلد برای جبران محدودیت ها و الگوریتم ژنتیک برای دستیابی به حداکثر عملکرد است. رویکرد مبتنی بر اصلاح GA، که genetic-fix نامیده می شود در [9] داده شده است که افراد با اندازه ثابت را تولید می کند و و اداره می کند تا بطور قابل ملاحظه ای فضای جستجو را کاهش دهد. الگوریتم ترکیبی tabu-greedy (ممنوع - حریمی) در [10] اجرا شده است. یک قاب TDMA با تعداد کمتر از اسلات های زمان، حداکثر تعداد انتقال با استفاده از روش جمعیت ممتاز و عملگر متقاطع اصلاح شده در الگوریتم ژنتیک در [12] ارائه شده است. حتی اگر راه حل بهینه در تعداد کمتری از نسلهها تشخیص داده شوند زمان اجرا کاهش نمی یابد. بر اساس مفاهیمی که برای فیلد ماشین حالت محدود ساخت شده در [14] داده شده است. ایستگاه هایی که می تواند بدون برخورد میان خودشان همه پخش کنند تحت عنوان حداکثر سازگاریها دسته بندی می شوند. باند پایین تر محکم ناشی شده از مجموعه ای از حداکثر سازگاریها که مبنایی برای استخراج حداقل طول قاب را تشکیل می دهد. الگوریتم مجموعه ای از قوانین را برای حداکثر سازگار به منظور به حداکثر رساندن استفاده از اسلات فراهم می کند. در [16] یک الگوریتم تصادفی ساده و سریع برای پیدا کردن یک استخر از راه حل های معتبر مشکل زمانبندی ارائه شده است. اگرچه با توجه به هر دو معیار توسط این الگوریتم زمان محاسبه کاهش نمی یابد. در [21] تدوین برنامه ریزی عدد صحیح خطی برای این مشکل پیشنهاد شده است، که در کاهش زمان اجرا انجام شده است اما حداکثر تعداد ایستگاه ها، در رویکرد در نظر گرفته شده 50 ایستگاه است. رویکرد تحمیل بی خردانه با برنامه نویسی پویا در [17] استفاده می شود به منظور بهبود کارایی با حذف موقعیت های تکرار شده و رویکرد الگوریتم ژنتیک هم تکاملی استفاده می شود برای حل مجموعه بدون برخورد برای شبکه های مش وایمکس. اشکال اصلی این الگوریتم هم تکاملی این است که هر عضو از جمعیت مورد آزمون با هر عضو جمعیت راه حل باید مقایسه شود. این، مقایسه ها و محاسبات زیادی نیاز دارد از این رو ممکن است هنگامی که اندازه جمعیت بسیار زیاد هستند سرعت این فرآیند پایین آید. شبکه عصبی شلوع بی نظم تکاملی (NCNN-G) برای حل مشکل زمانبندی همه

پخشی NP-complete در شبکه های رادیویی بسته ای در مرجع [22] داده شده است. رویکرد بهینه سازی دو مرحله ای برای رسیدن به دو هدف با دو تابع انرژی مختلف به تصویب رسید. در مرحله اول، یک G-NCNN که شبکه عصبی شلوغ بی نظم (NCNN) و طرح گسترش تدریجی را با هم ترکیب می کند تا طول فریم TDMA حداقل را پیدا کند. در مرحله دوم، NCNN مورد استفاده قرار گیرد تا حداکثر انتقال های گره در قاب TDMA که در مرحله اول به دست آمده است را پیدا کند. شبکه های عصبی شلوغ بی نظم هیستریک (HNCNN) [15] است با کنترل نویز یا اختلال مدل معادل پیشنهاد شده است. آنها HNCNN را با طرح گسترش تدریجی ترکیب می کنند تا طول قاب حداقل را در مرحله اول پیدا کنند، و انتقال بدون درگیری در فاز دوم را به حداکثر برسانند. در [23]، کوتاه ترین مسیر بر اساس طرح مسیریابی پروتکل اینترنت بار متعادل با مدل لوله (SLBIP)، ارائه شده است. شبکه ها با تعداد گره ها، لینک ها و درجه متوسط مختلف برای ارزیابی عملکرد در نظر گرفته شده است. زمان محاسبه برای حل مشکلات مسیریابی با الگوریتم های دیگر تجزیه و تحلیل شده تا خوبی این الگوریتم را نشان دهد. در مرجع [24]، یک شبکه عصبی بی نظم مورد استفاده قرار گرفته تا درخت مسیریابی چندپخشی تاخیر- محدود شده را محاسبه کند. دوازده شبکه مختلف با اندازه گره های متفاوت، لینک ها، گره مقصد و تاخیر باند متفاوت برای ارزیابی الگوریتم در نظر گرفته شده است. نتایج اساساً بر زمان محاسبات متمرکز شده است. یک الگوریتم توزیع شده ساده ای که هر دو ثبات و ذاتا تثبیت تحت یک مدل واقع بینانه است تا پیام های روی همه کوتاه ترین مسیر گسسته گره از یک فرایند به دیگری در یک شبکه مکعب N بعدی مسیریابی کند در مرجع [25] ارائه شده است. دنباله ای از اصل موضوع داده شده است تا زمان الگوریتم توزیع شده را تشخیص دهد. برای مشکل زمان بندی همه پخشی، روش های تجزیه و تحلیل شده در بالا از داد و ستد بین کیفیت راه حل و زمان اجرا رنج می برد. زمان محاسبات یک عامل مهم در اعتبار یک الگوریتم است. بنابراین یک الگوریتم هنوز برای مشکل زمانبندی همه پخشی نیاز دارد که کیفیت راه حل را در کاهش زمان محاسبه حتی برای یک شبکه بزرگ بهبود دهد. هدف از این کار این است که در یک زمان اجرای قابل قبول، زمان اسلات ها را کاهش دهد و تعداد کل انتقال ها را به حداکثر برساند. زمانبند WMN که در اینجا در نظر گرفته شده از نوع فعال ساز گره است علاوه بر این یک پرش و دو پرش با اطلاعات مورد نیاز اتصال کم می باشد. زمانبند، انتقال زمانبندی شده ای را مشخص می کند که استفاده از کانال را با تضمین QoS برای تمام MS به حداکثر می رساند. یک زمانبند با سه الگوریتم تکاملی اجرا می شود و یک سری شبیه سازی برای ارزیابی عملکرد MA پیشنهادی از نظر کیفیت راه حل و زمان اجرا، و به منظور بررسی برتری خود نسبت به الگوریتم های GA و IGA انجام می دهد. الگوریتم های ژنتیک به طور موثر بسیاری از مشکلات جستجو و بهینه سازی را حل می کند. با این حال، آنها ممکن است به راه حل های بهینه محلی انداخته شوند و یا آنها ممکن است راه حل بهینه ای را بوسیله سرعت همگرایی پایین پیدا کنند و الگوریتم GA کورکورانه در فضای جستجو سرگردان شود. برای غلبه بر این مشکلات، ما از مفهوم ایمنی استفاده کردیم تا به الگوریتم GA کمک کنیم. الگوریتم ژنتیک ایمنی اطلاعاتی از ماتریس پرش در طول واکنش سوسون فرایند بدست می آورد. الگوریتم IGA تعداد انتقال را در یک اسلات زمانی کاهش یافته افزایش می دهد، اما نه در یک زمان محاسبه خوب، MA زمان پردازش را کاهش می دهد. الگوریتم Memetic نسخه پیشرفته ای از الگوریتم تکاملی (EA) است. علاوه بر داروینسم، MA نظریه Lamarckian که فرزندان می توانند وارث دانش و یا ویژگیهایی که والدینشان در طول عمر خود به دست آوردند باشند، را قبول کردند. MA این ایده را با یکپارچه سازی ارتقاء محلی پیاده سازی کرده است، مانند جستجوی محلی و اپراتور تعمیر، به داخل EA استاندارد، و افزایش قابلیت ارث بری، این یکپارچگی به طور قابل توجهی توانایی بهره برداری از EA بهبود می بخشد. در الگوریتم ژنتیک، جهش ژنهای جدید برای مردم ایجاد می کند و عامل های متقاطع به جستجوی بهترین راه حل از ژن در مردم جهت می دهند. در الگوریتم memetic، این جهت گیری با جستجوی محلی به دست می آید. جستجو محلی، فضای جستجو را کاهش می دهد و سریع تر به راه حل با کیفیت بالا می رسد. MA به طور فعال بهبود راه حل را نشانده گرفته است و به صراحت با بهره برداری از تمام دانش موجود در مورد مشکل، اهمیت می دهد. بقیه این مقاله به شرح زیر سازماندهی شده است.

بخش 2 یک تعریف رسمی از این مشکل، همراه با محدودیت ها می دهد.

در بخش 3، ما سه الگوریتم و اپراتورهای آن را توصیف می کنیم.

جزئیات نتایج شبیه سازی، مقایسه اسلات زمانی، استفاده از کانال، متوسط تاخیر زمانی و زمان محاسبه شده توسط MA، IGA با دیگر الگوریتم های رقابتی در بخش 4 آمده است.

در نهایت، نتیجه گیری در بخش 5 به تصویر کشیده شده است.

2 معرفی زمانبند WMN

زمانبند WMN یک زمانبند بدون برخورد با حداقل طول فریم TDMA و حداکثر استفاده اسلات توسط گره ها، در یک زمان اجرای قابل قبول را تعیین کرده است. در زمانبند فرض بر این است که هر یک از MS اطلاعات مربوط به اتصال به شبکه در داخل شعاع دو پرش را دارد. در ابتدا، هر MS یک اسلات زمانی را در یک قاب اختصاص داده است (به عنوان مثال، MS، i بازه زمانی i ام اختصاص داده شده است). MS هایی که بیش از دو گره دورتر از MS i هستند نیز واجد شرایط انتقال در طول اسلات i هستند. برخی قانون از پیش تثبیت شده استفاده می شود تا یک MS واجد شرایط برای انتقال در اسلات i را انتخاب کند. MS انتخاب شده یک پیام همه پخش می فرستد تا به دیگر MS هایی که از اسلات i استفاده می کنند اطلاع دهد. الگوریتم به این روش پیشرفت می کند که به بسیاری از MS ها اجازه می دهد که امکان انتقال در هر اسلات را داشته باشند. یک گره ممکن است با گره های دیگر تداخل داشته باشد، بنابراین این گره ها نباید به طور همزمان انتقال داشته باشند. با این حال، تخصیص اسلات عادلانه میان همه MS را تضمین نمی کند و توپولوژی شفاف نیست. WMN می تواند توسط گراف بدون جهت $G = (N, E)$ که در آن N مجموعه ای از گره ها و E نشان داده لینک ها (انتقال) است که دو طرفه فرض شده است. $|N|$ نشان دهنده تعداد گره ها در شبکه داده شده بعنوان مثال $|N| = \{n_1, n_2, \dots, n_x\}$ ، و $|M|$ تعداد اسلات های زمانی است. در شکل یک $N = \{A, B, C, D, E, F, G, H\}$ و $|N| = 8$ است. ماتریس اتصال [CM] نشان دهنده یک ارتباط مستقیم بین گره ها، ماتریس پرش [HM] در مورد اطلاعات اتصال یک پرش و دو پرش هر گره می گوید، ماتریس زمانبند [SM] اسلات های زمانی اختصاص داده شده به شبکه داده شده بدون هیچگونه تداخل است.

$$[CM] = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

این ماتریس اتصال برای شبکه داده شده در شکل 1 تعریف شده است. ستون نشان دهنده گره های شبکه و سطر نشان دهنده ارتباط موجود بین گره ها است، به عنوان مثال، ردیف یک در مورد اطلاعات اتصال گره A می گوید، به همین ترتیب برای گره های باقی مانده. ماتریس مقدار 0 یا 1 را دارد، که 1 نشان دهنده وجود یک لینک می باشد.

$$[PM] = \begin{matrix} & \begin{matrix} A & B & C & D & E & F & G & H \end{matrix} \\ \begin{matrix} D \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{matrix} & \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{matrix}$$

این ماتریس پرش برای شبکه داده شده در شکل 1 تعریف شده است. مقدار ردیف نشان دهنده اطلاعات یک پرش و دوپرش بین گره ها است.

$$[SM] = \begin{matrix} & \begin{matrix} A & B & C & D & E & F & G & H \end{matrix} \\ \begin{matrix} 1 \\ 0 \\ 0 \\ 0 \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \end{matrix}$$

ماتریس زمانبند TDMA بهینه برای شبکه در شکل 1 تولید شده است. سطر تعداد اسلاتهای زمانی را نشان می دهد. آن مقدار 0 یا 1 را می گیرد، که در آن 1 نشان می دهد گره اجازه انتقال در آن بازه زمانی را دارد. در اسلات زمانی اول، گره های A و E اجازه دارند بسته های خود را بدون تداخل ارسال کنند.

3 الگوریتم های تکاملی

الگوریتم ژنتیک یا GA یک تکنیک جستجوی ابتکاری است که فرایندهای انتخاب طبیعی و تکاملی را شبیه سازی می کند [26]. جان هلند برای اولین بار الگوریتم ژنتیک (GA) در سال 1960 ارائه داده است. GA ها، روش جستجوی موثر و قوی برای مسائل NP-complete هستند [27]. انتخاب، تقاطع، جهش، تابع صلاحیت و شرط خاتمه در زیر برای هر سه الگوریتم متداول بحث شده است. ماتریس زمانبند TDMA یک ماتریس $M * N$ است که در آن M تعداد اسلات زمانی است و N تعداد کل گره ها در شبکه است. ماتریس زمانبند به عنوان کروموزوم رشته بیتی شامل 0ها و 1ها نشان داده شده است. هر سطر و ستون از ماتریس زمانبند یک اسلات زمانی و انتقال گره را نشان می دهد. مقدار 1 در موقعیت (i, j) در ماتریس نشان می دهد که گره j ام برای انتقال در اسلات زمانی i ام مجاز است. فریم اولیه TDMA با استفاده از روش جمعیت برگزیدگان چاکرابورتی ساخته شده است [12]. GA، IGA و MA مجموعه ای از کروموزوم های دست کاری شده اند تا یک راه حل بهینه را جستجو کنند. عامل های انتخاب برای انتخاب والدین و انتخاب بازمانده، اصل بقای اصلح داروینی را دنبال می کنند. اول، انتخاب والدین برای فرایند تولید مثل است، معمولاً یک توضیح جایگزین انتخاب طبیعی، بعنوان مثال مشخصه افراد باید احتمال بیشتر تولید مثل را داشته باشد. این مطالعه K-مسابقه انتخابی برای انتخاب والدین اجرا می کند، برنده را در میان K فرد از جمعیت به طور تصادفی انتخاب می کند. K تعداد، فشار انتخاب را کنترل می کنند، به K بالاتر فشار انتخاب بالاتر می دهد. دوم، انتخاب بازمانده اصلح بازمانده اصلح را فراهم می کند. فقط افراد اصلح به عنوان والدین نسل بعدی انتخاب شده اند. ایده نخبه گرایی برای حفظ تعدادی از بهترین افراد در هر نسل استفاده می شود. در این مقاله، درصد کمی از مناسبترین افراد برای نسل بعدی باقی می ماند. این کارایی الگوریتم را با جلوگیری از دست رفتن بهترین راه حل افزایش می دهد. از هر نسل 10٪ از بهترین راه حل ها برای تکرار بعدی باقی می ماند. کروموزوم های انتخاب شده برای تولید مثل در استخر جفت گیری جمع شده بودند. اپراتور تقاطع تک نقطه ای روی ردیفی از جمعیت انجام شده است. هنگامی که یک نقطه تقاطع مشخص شده است، یک ردیف تصادفی از والدین اولین PR1 با یک سطر تصادفی از والدین دوم PR2 روبرو می شود. CH1 کروموزوم حاصل با PR1 جایگزین شده است و CH2 با PR2 جایگزین شده

است. پس از جایگزینی، اگر راه حل محدودیت ها را نقض کند پس آن جریمه شده است. عملگر جهش به شیوه ای متفاوت بسته به تناسب ژن انتخابی رفتار می کند. عملگر جهش یک بیت در کروموزوم انتخاب شده را با توجه به صلاحیت شخص تغییر می دهد. تابع صلاحیت کیفیت (صلاحیت) راه حل های نامزد را ارزیابی می کند. تابع صلاحیت برای مشکل زمانبندی مبتنی بر دستیابی کانال متغیر و باند تنگ پایینی تر است. نقطه پایان تعیین می کند که آیا بهترین راه حل در آن نسل شناسایی شده است یا نه. بهترین راه حل عملی، این است که هر دو معیار قانع شوند یا راضی شوند. زمانی که نسل تکامل به این نقطه پایان می رسد، الگوریتم متوقف می شود و راه حل بهینه برای شبکه داده شده خارج می شود.

3.1 الگوریتم ژنتیک

پس از مقدار دهی اولیه جمعیت، عملگر انتخاب دو کروموزوم را از جمعیتی که به عنوان والدین به خدمت گرفته شده اند، برمی دارد. سپس اپراتور متقاطع اطلاعات را بین این دو والدین مبادله می کند تا فرزندان خود را تولید کنند. نرخ متقاطع از پیش تعیین شده، احتمال اجرای متقاطع را تعریف می کند. جهش با یک احتمال، به نام نرخ جهش، برای تغییر به آرامی برخی از ژن ها در فرزندان انجام می شود.

الگوریتم 1 چارچوب الگوریتم ژنتیک را نشان می دهد.

Algorithm 1 Genetic algorithm

```

initialize population GAPop;
evaluate GAPop;
while (not terminated)
{
    GAPs = Select (GAPop);
    GAPc = Crossover (GAPs);
    GAPm = Mutate (GAPc);
    GAP' = evaluate GAPm;
    GAPop = Survival (GAPop, GAP');
}

```

جمعیت تولید شده با شرایط صلاحیت مورد ارزیابی قرار می گیرد. اگر راه حل بهینه در تولید تعریف شده باشد پس الگوریتم با راه حل خاتمه یافته است، وگرنه روش نخبه گرایی بر روی جمعیت انجام می شود و به تولید بعدی می پردازد. در پایان تکرار، جمعیت های تولید شده در نسل برای حذف ردیف تکراری گرفته می شوند یعنی، بازه زمانی که تکرار شده است به منظور تولید قاب TDMA بهینه از جمعیت حذف می شود.

3.2 الگوریتم ژنتیک ایمنی

در GA دو عامل اصلی ژنتیکی متقاطع و جهش، اگر چه به هر فرد شانس تکاملی برای به دست آوردن مطلوب جهانی را می دهند اما همچنین به دلیل جستجو های تصادفی و بدون نظارت در طول پروسه کامل تا حدی باعث انحطاط می شود. از سوی دیگر، GA توانایی استفاده از برخی از ویژگی های اساسی و آشکار و یا دانش مشکل معلق را ندارد. بر اساس ملاحظات فوق، الگوریتم ژنتیک ایمنی ارائه شد.

الگوریتم 2 ساختار الگوریتم ژنتیک ایمنی را نشان می دهد.

Algorithm 2 Immune genetic algorithm

```
initialize population IGAPop;  
evaluate IGAPop;  
while (not terminated)  
{  
    IGAPs = Select (IGAPop);  
    IGAPc = Crossover (IGAPs);  
    IGAPm = Mutate (IGAPc);  
    Immunization (IGAPm)  
    {  
        IGAPsel = ImmuneSelection (IGAPm);  
        IGAPv = Vaccination (IGAPsel);  
    };  
    IGAP' = evaluate IGAPv;  
    IGAPop = Survival (IGAPop, IGAP');  
};
```

راه حل بعد از مرحله تولید مجدد برای عملیات ایمنی اتفاق می افتد. IGA یک الگوریتم بهینه سازی هوشمند است که به طور عمده یک اپراتور ایمنی ساخته شده است و توسط دو مرحله انجام شده است. : انتخاب ایمنی و واکسیناسیون
دانش اضافه شده به الگوریتم IGA به صورت زیر اجرا کرده است.

3.2.1 انتخاب ایمنی

جمعیت به تازگی ایجاد شده پس از تولید مثل، که محدودیت های اولیه و ثانویه، ارضا می کند برای حذف ردیف تکراری انتخاب شده است. جمعیت به دست آمده با توجه به دستیابی به کانال متغیر و ذخیره شده در استخر واکسن مرتب می شوند.

3.2.2 واکسیناسیون

واکسیناسیون برای بهبود صلاحیت با تغییر ژن یک جمعیت فردی با دانش قبلی برای به دست آوردن صلاحیت بالاتر با احتمال بیشتری استفاده می شود. یک کروموزوم از استخر واکسن برای واکسیناسیون گرفته شده است. IGA اولین انتقال گره در جامعه را شناسایی می کند. در طول همان اسلات زمانی، برخی از گره های دیگر، که تداخلی با گره انتقال ندارند، می توانند اجازه انتقال در همان بازه زمانی را داشته باشند. برای انجام این کار، یک گره به صورت تصادفی انتخاب شده است و با ماتریس پرش چک شده است که آیا آن در حال حاضر با گره انتقال ایجاد تداخل می کند، اگر نه ارزش گره به یک جهش یافته، به گره انتخاب شده در همان بازه زمانی اجازه انتقال می دهد. ژن کروموزوم انتخاب شده بر اساس دانش به دست آمده از ماتریس پرش شبکه داده شده ویرایش می شود از این رو فرایند واکسیناسیون تعداد انتقال ها را افزایش می دهد.

3.3 الگوریتم Memetic

الگوریتم های Memetic (MA) تعمیم الگوریتم های تکاملی (EA) است که فرآیندهای جستجوی محلی در عوامل و تلاش برای بهبود صلاحیتشان را فراهم کرده است [28-31]. در مقایسه با روش های دیگر، الگوریتم های Memetic، به دلیل کاربرد گسترده ای برتر است.

با وجود نتایج خوبی به دست آمده توسط برخی از MA، روند طراحی کارآمد MA اغلب به جزئیات مشکل خاص بستگی دارد.

ساختار الگوریتم Memetic در الگوریتم 3 نشان داده شده است.

Algorithm 3 Memetic algorithm

```

initialize population MPop;
evaluate MPop;
while (not terminated)
{
     $MP_s = \text{Select}(MPop)$ ;
     $MP_c = \text{Crossover}(MP_s)$ ;
     $MP_m = \text{Mutate}(MP_c)$ ;
    MemeticAlg( $MP_m$ )
    {
         $MP_{op} = \text{Optimizer}(MP_m)$ ;
         $MP_l = \text{Improver}(MP_{op})$ ;
    }
     $MP' = \text{evaluate } MP_l$ ;
     $MPop = \text{Survival}(MPop, MP')$ ;
}

```

جمعیت اولیه با استفاده از روش جمعیت نخبگان و انتخاب والدین برای تولید مثل است و با استفاده از انتخاب K-مسابقات انجام می شود ساخته شده است. روی کروموزوم های انتخاب شده، یک اپراتور متقاطع تک نقطه ای اجرا شده است و عملگر جهش بر اساس جهش احتمال داده شده انجام شده است. بعد از تقاطع و جهش، دنباله بهینه ساز و اصلاح کننده بر روی کروموزوم ها در MA اجرا شده است شود.

3.3.1 بهینه ساز

مرحله بهینه ساز MA، تعداد شکاف های زمانی را بوسیله تعیین کانال دستیابی برای هر گره کاهش می دهد. qx عملکرد گره X در جمعیت جاری است، به عنوان مثال، تعداد کل انتقال ها اجرا شده توسط نود X در اسلات زمان داده شده، با استفاده از Eq تعریف شده است. (2). فاز بهینه ساز انتقالهای هر گره را بدست می آورد، سپس تعریف می کند که آیا همان گره در برخی از دیگر اسلات زمان L در حال انجام انتقال است. در بازه زمانی J ام، اگر گره هایی که منتقل شده اند شامل $1 < qx$ پس ردیف از آن جامعه حذف شده است. در [12] ردیف هایی که زیر مجموعه ای از ردیف تولید شده پس از متقاطع هستند، حذف شده اند. در این مقاله، حذف ردیف تکراری در GA و IGA، کاهش اسلات های زمانی را شکل داد، اگر یک اسلات زمانی در آن جمعیت تکرار شده است، در حالی که در MA بهینه ساز کاهش را بر اساس ارزش qx اجرا کرده است. این مرحله جمعیتی را تولید می کند با حداقل تعداد اسلات زمانی با محدودیت که هر گره حداقل یک بار در آن فریم TDMA منتقل شده است.

3.3.2 اصلاح کننده

بهبودساز یک راه حریصانه است و این تنوع راه حل را کاهش می دهد. جمعیت هایی از بهینه ساز برای مرحله بهبودساز گرفته شده اند جایی که تعداد کل انتقال ها در اسلات های زمانی کاهش یافته، افزایش یافته است. دانش بدست آمده از ماتریس پرش، مرحله بهبودساز تعداد انتقال های پس از کاهش اسلاتهای زمانی را افزایش می دهد. از آنجا که عملیات الگوریتم memetic در هر تکرار به اجرا درآمد، راه حل بهینه در تعداد کمتری

از نسل ها تعریف شده است. از این رو، زمان اجرای الگوریتم نیز در مقایسه با سایر الگوریتم های رقابتی که اخیرا پیشنهاد شده است، کاهش یافته است.

4 نتایج شبیه سازی

مجموعه ای از شبیه سازی های برای ارزیابی عملکرد MA پیشنهادی انجام شده است تا مشکل زمانبندی همه پخشی را حل کند، در مقایسه با بازپخت فیلد میانگین [3]، GA [12] و الگوریتم ژنتیک جایگشت با صالحیت [13].

بخش زیر بحث در مورد نتایج شبیه سازی در مورد تعداد گره $|N|$ ، تعداد اسلات های زمانی $|M|$ و درجه شبکه ها است. عوامل تابع صلاحیت تعریف شده است به عنوان،

1. متغیر دستیابی به کانال

برای کل شبکه:

$$\sigma = \frac{1}{|M| * |N|} \left(\sum_{i=1}^{|M|} \sum_{j=1}^{|N|} [SM_{ij}] \right) \quad (1)$$

و برای هر نود:

$$\rho_x = \sum_{i=1}^{|M|} [SM_{ix}] \quad (2)$$

$$\sigma_x = \frac{\rho_x}{|M|} \quad (3)$$

باند پایین تر تنگ

$$ND = \max_{n \in N} |D(n)| \quad (4)$$

$D(n)$ مجموعه درجه گره های n است، ND نشان دهنده حداکثر درجه شبکه است، بر اساس این مقدار باند تنگ پایین تر تولید شده است به عنوان

$$\Delta = |M| - ND \geq 1 \quad (5)$$

اگر $D = 1$ پس راه حل بهینه است. شرایط ترمینالی برای سه الگوریتم در این مطالعه بحث شده است، $D = 1$ یا حداکثر تعداد نسل ها که در تمام آزمایش های ما 500 گرفته شده است.

شبکه های مختلف تولید شده به طور تصادفی با درجه و گره های مختلف، سه الگوریتم را امتحان کرده که هر کدام یک توپولوژی multihop را نشان می دهد. برای تنظیمات خاصی از پارامترها، الگوریتم 150 بار انجام شده، مقدار متوسط نتایج در نتایج شبیه سازی زیر داده شده است.

نتایج این شبیه سازی بر اساس پارامترهای اندازه جمعیت 50، حداکثر تعداد نسل 500، نرخ متقاطع 0.30، احتمال جهش 0.001 است و بر روی این سه معیار،

1. باند پایین تر تنگ مقدار D یک است.

2. متغیر دستیابی به کانال برای پیدا کردن بهبود در تعداد انتقال.

3. زمان اجرا بر روی یک پلت فرم شبیه سازی است با استفاده از کد Matlab فعال بر روی ویندوز XP با مشخصات سیستم Intel Core 2 Duo T6600 2.2 GHz

4.1 نتیجه شبیه سازی GA

هدف از اولین شبیه سازی این بود که عملکرد الگوریتم ژنتیک را برای شبکه های مختلفی که در جدول 1 نشان داده شده بررسی کنیم.

Table 1 Simulation results of genetic algorithm

No. of nodes	No. of links	Average degree	Avg. ND	Maximum ND	Minimum TDMA frame length	Avg. σ	Avg. no. of generations	Computation time
15	25	3.3	4	6	7	0.219	30.6	0.80 s
30	49	3.3	4.8	8	9	0.156	27.2	01.10 min
80	156	3.9	5.8	8	9	0.154	238	16.08 min
100	200	4	7.5	9	10	0.104	422	32.00 min

تعداد گره ها برای شبیه سازی از محدوده پنج تا صد گرفته است. شبکه های گره کوچکتر با تعداد بیشتری انتقال در نسل قابل قبول اجرا می شود. با این حال، برای شناسایی یک شبکه 100 گره با 200 لبه با درجه نه راه حل بهینه قاب TDMA پس از 489 نسل شناسایی می کند. متوسط تعداد نسل برای شبکه 100 گره ای 422 می باشد. این به منظور کاهش زمان اجرا باید کاهش یابد.

4.2 نتیجه شبیه سازی IGA

جدول 2 نشان دهنده خروجی تولید شده توسط الگوریتم ژنتیک ایمنی برای تعدادی از گره ها و لبه های مختلف است.

Table 2 Simulation results of immune genetic algorithm

No. of nodes	No. of links	Average degree	Avg. ND	Maximum ND	Minimum TDMA frame length	Avg. σ	Avg. no. of generations	Computation time
15	25	3.3	4	6	7	0.289	18.6	0.5 s
30	60	4	4	8	9	0.199	21.0	12 s
40	80	4	6	8	9	0.187	35.2	3.8 s
50	100	4	6	8	9	0.180	49.7	10.23 s
70	140	4	7	8	9	0.172	64.9	2.49 min
80	160	4	7	8	9	0.167	89.0	4.0 min
100	200	4	7	9	10	0.141	92.3	12.61 min
100	250	5	8	9	10	0.117	98.0	27.37 min

در مقایسه با الگوریتم ژنتیک، دانش اضافه شده IGA می تواند توانایی جستجو و سازگاری را بهبود بخشد، تا حد زیادی سرعت همگرا را افزایش دهد [32-34]. در طول فرآیند واکسناسیون، آنتی ژن انتخاب شده است با تعداد بیشتری انتقال بهبود یافته است به طوری که دستیابی به کانال افزایش می یابد. در مقایسه نتایج شبیه سازی IGA در جدول 2 با GA در جدول 1، تعداد نسل ها کاهش می یابد، و تعداد متوسط انتقال هر شبکه بهبود یافته است. برای شبکه ای با 80 گره و 100 گره، راه حل با نسل قابل قبول تعریف شده است. برای شبکه های 100 گره ای با درجه متوسط چهار و پنج، راه حل بهینه در 16 دقیقه و 25 دقیقه تعریف شده است. با این حال، اولین دو معیار توسط IGA راضی کننده است، اما سومی، به عنوان مثال، زمان اجرا برای یک شبکه بزرگ کاهش نمی یابد.

4.3 نتیجه شبیه سازی MA

روش هایی تا کنون مورد بحث قرار گرفته عمدتاً در همگرایی الگوریتم ها در شرایط باند پایین تر تنگ و افزایش تعداد انتقال ها متمرکز شده است.

بنابراین، این پرسش مطرح می شود: رابطه این روش ها در مقایسه با یکدیگر از نظر زمان چیست ؟

این به عنوان سوال اصلی الگوریتم Memetic تنظیم شده است. نتیجه شبیه سازی MA برای پیدا کردن بهره وری ارزیابی شد، به عنوان مثال، سرعت همگرایی. از جدول 1 و 2 روشن است که، IGA بهبود داده است دستیابی به کانال را در تعداد نسل های کاهش یافته نیز در زمان محاسبه کمتر در مقایسه با GA. با دستیابی به کانال متوسط، متوسط تعداد نسل و زمان محاسبات شبکه های مختلف با استفاده از MA برای شبکه ای با درجه های مختلف در جدول 3 تجزیه و تحلیل شده است.

Table 3 Simulation results of memetic algorithm

No. of nodes	No. of links	Average degree	Avg. ND	Maximum ND	Minimum TDMA frame length	Avg. σ	Avg. no. of generations	Computation time
50	85	3.4	4	6	7	0.203	5.12	7 s
80	152	3.8	7	9	10	0.175	7.48	11 s
100	150	3	6.9	9	10	0.170	9.03	1.7 min
100	200	4	7.5	10	11	0.162	16.58	2.0 min
100	250	5	7	10	11	0.147	17.0	2.0 min
100	300	6	7	8	9	0.121	19.76	2.3 min
200	400	4	8	10	11	0.150	29.76	12.2 min
200	500	5	8	9	10	0.139	38.02	20.8 min
300	600	4	7	10	11	0.176	55.54	30.5 min
400	800	4	12	16	17	0.159	60.58	65.3 min
500	1,000	4	9	14	15	0.128	89.04	72.11 min

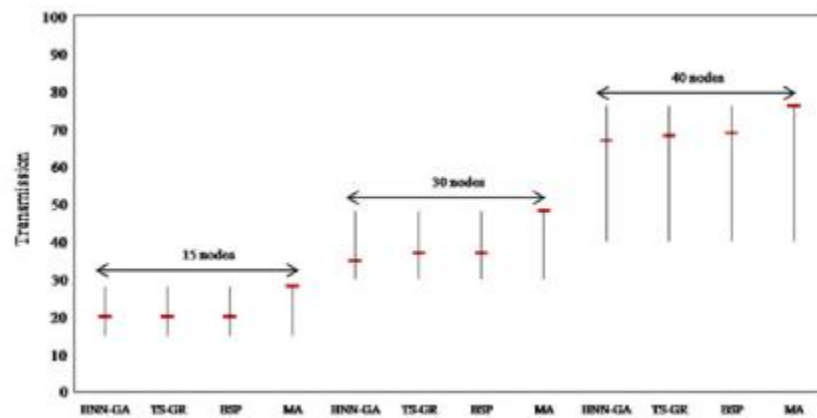
زمان برای شبکه 100 گره ای در MA 2.0 دقیقه است، کارآمد تر از زمان صرف شده توسط IGA برای همان شبکه در 12.61 دقیقه است. برای شبکه ای با بیش از 100 گره، زمان محاسبه در IGA و سایر روش های کارآمد به تازگی پیشنهاد شده هنگامی که ما با MA مقایسه می کنیم بسیار کارآمد نیستند. این مزیت اصلی MA است. تعداد کل انتقال های تولید شده توسط MA برای شبکه های گره های مختلف با اسلات های زمانی متفاوت با هم مقایسه شده GA-HNN [8]، GR-TS [10] و BSP [21] و در جدول 4 نشان داده شده است.

Table 4 Comparison of MA with other familiar algorithms in terms of number of transmissions

No. of nodes	Time slots (MI)	HNN GA [8]	TS GR [10]	BSP [21]	MA
15	8	20	20	20	28
30	10	35	37	37	48
30	11	40	43	-	51
30	12	47	48	-	54
40	8	67	68	69	76
40	9	77	77	-	84

نمودار مقایسه انتقال MA با این الگوریتم ها نیز در شکل 2 نشان داده شده است..

Fig. 2 Transmission comparison graph of existing algorithm with MA



نقطه شروع خط عمودی در نمودار نشان دهنده کمترین مقدار انتقال برای شبکه داده شده است، نقطه پایان خط نشان دهنده بالاترین مقدار انتقال و یک خط افقی کوچک نشان دهنده تعداد انتقال های تولید شده توسط هر الگوریتم است. جدول و شکل ثابت می کند که MA تعداد بیشتری انتقال را برای شبکه های مختلف تولید می کند در مقایسه با الگوریتم های موجود و با تفاوت 6-11 تا در مجموع تعداد انتقال ها تولید می کند. زمان محاسبه و تعداد نسل ها برای شناسایی راه حل های مطلوب کاهش می یابد در جایی که دستیابی به کانال در MA افزایش یافته در مقایسه با GA [12]. مقایسه متوسط زمانی گرفته شده توسط MA و تغییر در الگوریتم ژنتیک [12] در شکل 3 داده شده است..

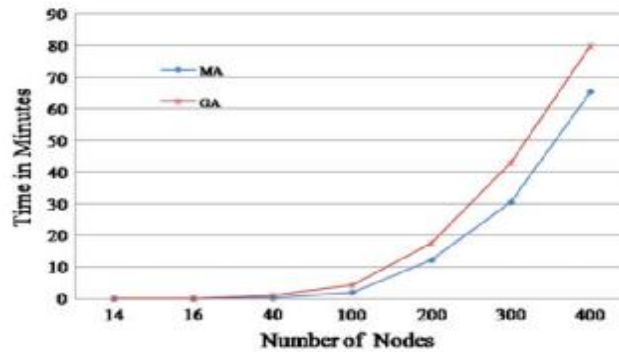


Fig. 3 Comparison of computation time taken by MA and GA [12]

دو مشکل benchmark در [3] مورد بحث قرار گرفته با استفاده از MA حل شده است و نتایج به دست آمده در مقایسه با الگوریتم های دیگر مانند، شبکه عصبی بی نظم شلوغ هیسترتیک تدریجی HNCNN-G [15]، شبکه های عصبی بی نظم شلوغ تدریجی NCNN-G [22]، الگوریتم های ژنتیکی هم تکاملی برای مجموعه بدون برخورد GACFS [17]، الگوریتم مبتنی بر ماشین حالت محدود FSMA [14]، الگوریتم ژنتیکی جایگشت با صلاحیت CPGA [13] و به الگوریتم فیلد اصلی بازپخت MFA [3] در جدول 5 نشان داده شده است.

Table 5 Comparison of MA with other competitive algorithms in terms of time slot lMl , channel utilization σ and time delay

Instance	Parameter	MA	G-HNCNN	G-NCNN	GACFS	FSMA	CPGA	MFA
#1	lMl	10	10	10	10	10	10	12
	σ	0.24	0.1233	-	0.093	0.1167	0.1233	0.1056
	Time delay	6.1529	8.83	9.0	-	9.2	-	10.5
#2	lMl	8	8	8	8	8	8	9
	σ	0.2844	0.2125	-	0.203	0.200	0.200	0.197
	Time delay	5.0433	5.7056	5.8	-	6	-	6.9

30 گره با 70 لبه در مثال # 1 تجزیه و تحلیل شده است و 40 گره با 66 لبه در مثال # تجزیه و تحلیل شده است 2 با در نظر گرفتن تعداد اسلات های زمانی، دستیابی به کانال و تاخیر زمانی. همانطور که در جدول 5 دیده می شود تاخیر زمانی محاسبه شده است، MA دستیابی به کانال را با کمترین تاخیر زمانی افزایش می دهد این نشان می دهد که MA در مقایسه با دیگر الگوریتم هایی که اخیرا پیشنهاد شده به طور موثرتری اجرا می شود.

شکل 4 تاخیر زمانی و دستیابی به کانال الگوریتم های مختلف را با MA مقایسه می کند.

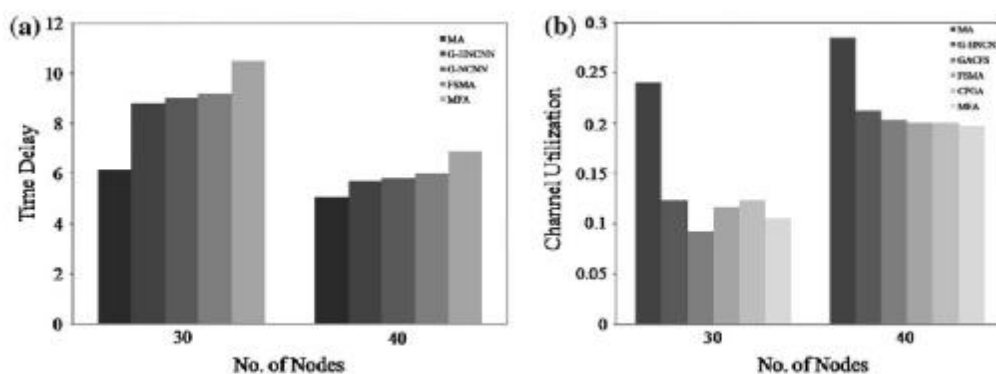


Fig. 4 a Comparison of average time delay, b comparison of channel utilization

جدول 6 زمان محاسبه MA را با الگوریتم های GA-HNN [8] و BSP [21] مقایسه کرده است و نشان می دهد که زمان محاسبه تا حد زیادی کاهش یافته است.

Table 6 Comparison of MA with other competitive algorithms in terms of computation time

No. of nodes	Average degree	HNN-GA [8] (s)	BSP [21] (s)	MA (s)
10	3	5.1	0.01	0.01
20	3	12.97	0.06	0.01
30	3	85.4	0.07	0.02
40	3	165.12	0.3	0.14
50	3	194.4	0.64	0.27
10	4	5.24	0.01	0.01
20	4	20.04	0.33	0.01
30	4	152.06	1.45	2.04
40	4	280.37	4.74	3.41
50	4	320.06	14.2	7.00

شکل 5(a) تعداد انتقال تولید شده توسط GA، IGA و MA با تعداد کل گره ها را مقایسه می کند.

شکل 5(b) تعداد نسل ها گرفته شده توسط GA، IGA و MA با تعداد گره های شبکه های مختلف مقایسه می کند.

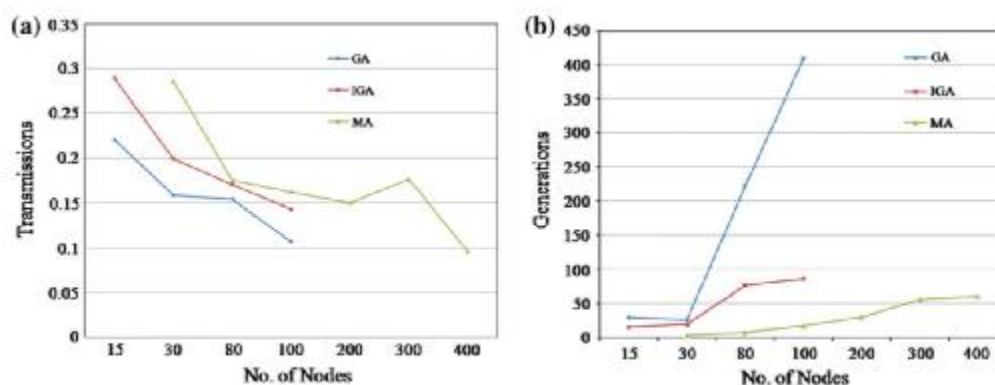


Fig. 5 a Comparison of total transmissions, b comparison of generations

این نتایج نشان می دهد که الگوریتم memetic در شرایط باند پایین تر تنگ بطور موثری اجرا می شود و دستیابی به کانال در زمان محاسبه قابل قبول افزایش می یابد. متوسط تاخیر زمانی گرفته شده توسط GA-HNN [8]، NCNN-G [22] و MA برای شبکه های مختلف از 15 تا 250 در شکل 6 مقایسه شده است.

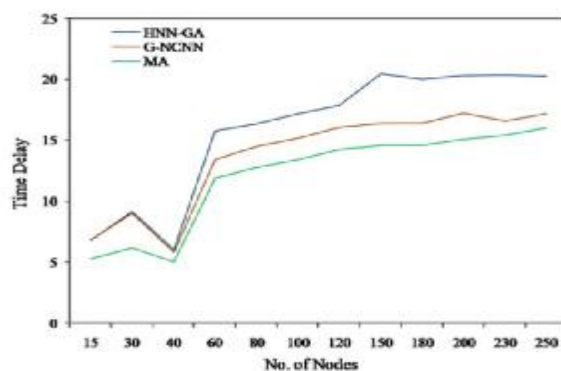
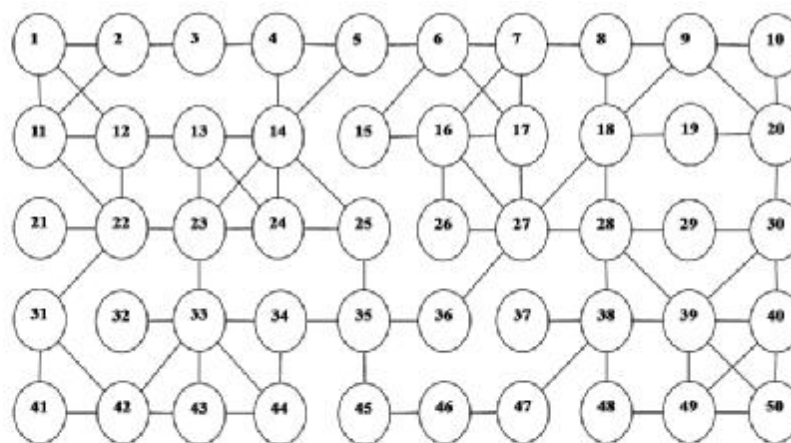


Fig. 6 Comparison of average time delay

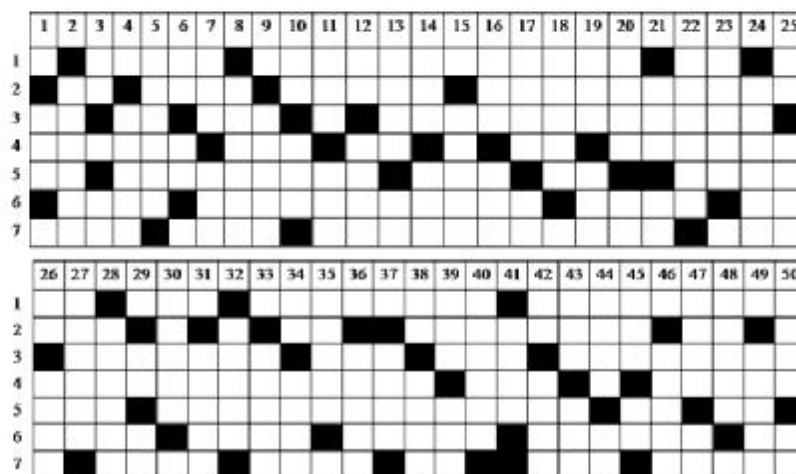
اگر تعداد انتقال بیشتری وجود داشته باشد کاهش تاخیر زمانی بوجود می آید، از شکل 6 همچنین بدست آمده که تعداد کل انتقال تولید شده توسط MA هنگامی که نسبت به دو الگوریتم دیگر مقایسه می شود بالا است.

یک نمونه شبکه 50 گره ای با 85 لبه با درجه متوسط 3.4 و قاب TDMA بهینه تعریف شده متناظرش که از MA استفاده می کند در شکل 7(a, b) نشان داده شده است. از آنجا که حداکثر درجه شبکه شش است MA زمانبند بهینه را در هفت اسلات زمانی با 61 انتقال تولید کرده است. دستیابی به کانال برای این شبکه 0.174 است و متوسط تاخیر زمانی 6.27 است.

Fig. 7 a Network with 50 nodes, 85 edges and the average degree of 3.4, b optimal solution found by MA for the same network



(a)



(b)

در جدول 7، زمان محاسبات MA برای تعدادی از گره ها و لینک های مختلف با SLBIP مقایسه شده است [23].

Table 7 Comparison of MA with SLBIP [23] in terms of computation time

No. of nodes	No. of links	SLBIP [23] (s)	MA (s)
6	24	0.031	0.011
12	36	1.015	0.073
12	48	3.119	1.492
15	56	16.558	8.961
20	68	6.289	2.133
35	100	1.613	0.918

از تمام جداول و ارقام متوجه می شویم که MA نه تنها در مقایسه با GA و IGA، بلکه نسبت الگوریتم هایی که اخیرا پیشنهاد شده و جلوتر در موردشان بحث شد اجرای بهتری دارد.

5 نتیجه گیری

الگوریتم ژنتیک پایه، دانش اضافه شده به الگوریتم ژنتیک ایمنی و یک الگوریتم memetic خاص دامنه مورد بحث قرار گرفت تا زمانبندی همه پخش شبکه های multihop بی سیم را حل کند. در مقایسه GA و IGA و MA به طور فعالی بهبود راه حل های را نشانه گرفته بود، در حالی که GA کورکورانه در فضای جستجو سرگردان بود. MA از تمام دانش موجود در مورد مشکل بهره برداری کرد، در حالی که الگوریتم ژنتیک ایمنی از دانش ماتریس پرش در طول فرایند واکسیناسیون استفاده کرد. IGA تعداد انتقال را در یک اسلات زمانی کاهش یافته افزایش می دهد اما زمان محاسبه خوب ندارد، MA بر آن غلبه می کند. در مقالات قبلی، اشکال اصلی به نقل از نویسندگان، زمان محاسبه برای شبکه های بزرگ است که تا حد زیادی در این مقاله کاهش می یابد. نتایج شبیه سازی مزایای MA را در شرایط استفاده از کانال، تعداد نسل ها، و زمان اجرا تایید می کند. MA به باند پایین تر تنگ در مقایسه با الگوریتم های دیگر در زمان اجرا کوتاه تر دست یافت. نتیجه اثربخشی و بهره وری MA را برای مشکل زمانبندی همه پخش تایید می کند. افزایش بیشتر می تواند انجام شود تا زمان محاسبات را حتی برای شبکه های بزرگ با بیش از 500 گره کاهش دهد.

مراجع

1. Lloyd, E. L. (2002). Broadcast scheduling for TDMA in wireless multihop networks. In Handbook of wireless networks and mobile computing (pp. 347–370). New York: Wiley.
2. Ephremides, A., & Truong, T. V. (1990). Scheduling broadcasts in multiple radio networks. IEEE Transactions on Communications, 38(4), 456–460.
3. Wang, G., & Ansari, N. (1997). Optimal broadcast scheduling in packet radio networks using mean filed annealing. IEEE Journal on Selected Areas in Communications, 15(2), 250–260.

4. Ju, J., & Li, V. O. K. (1998). An optimal topology-transparent scheduling method in multihop packet radio networks. *IEEE/*

ACM Transactions Networking, 6(3), 298–306.

5. Wang, C.-Y., Atkinson, M. W., Fertig, K. W., & Sastry, A. R. K. (1986). Performance evaluation of multi-hop packet radio networks using simulation. In *MILCOM '86—Military Communications Conference*, Monterey, CA, October 1986, pp. 28.5.1–28.5.6.

6. Funabiki, N., & Takefuji, Y. (1993). A parallel algorithm for broadcast scheduling problems in packet radio networks. *IEEE*

Transactions on Communications, 41(6), 828–831.

7. Goutam, C., & Hirano, Y. (1998). Genetic algorithm for broadcast scheduling in packet radio networks. In *Proceedings of IEEE*

World Congress Computational Intelligence, Anchorage, AK, May 1998, pp. 183–188.

8. Salcedo-Sanz, S., Bousoño-Calzon, C., & Figueiras-Vidal, A. R. (2003). A mixed neural-genetic algorithm for the broadcast

scheduling problem. *IEEE Transactions on Wireless Communications*, 2(2), 277–283.

9. Ngo, C. Y., & Li, V. O. K. (2003). Centralized broadcast scheduling in packet radio networks via genetic-fix algorithms.

IEEE Transactions on Communications, 51(9), 1439–1441.

10. Peng, Y., Soong, B. H., & Wang, L. (2004). Broadcast scheduling in packet radio networks using mixed tabu-greedy algorithm.

Electronics Letters, 40(6), 375–376.

11. Shi, H., & Wang, L. (2005). Broadcast scheduling in wireless multihop networks using a neural-network-based hybrid algorithm. *Neural Networks*, 18, 765–771

12. Chakraborty, G. (2004). Genetic Algorithm to solve optimum TDMA transmission schedule in broadcast packet radio networks.

IEEE Transactions on Communications, 52(5), 765–777.

13. Wu, X., Sharif, B. S., Hinton, O. P., & Tsimenidis, C. C. (2005). Solving optimum TDMA broadcast scheduling in mobile ad hoc networks: A competent permutation genetic algorithm approach. *IEE Proceedings: Communications*, 152(6), 780–788.

14. Ahmad, I., Al-Kazemi, B., & Das, A. S. (2008). An efficient algorithm to find broadcast schedule in ad hoc TDMA networks. *Journal of Computer Systems, Networks, and Communications*, 12, 1–10.

15. Sun, M., Zhao, L., Cao, W., Xu, Y., Dai, X., & Wang, X. (2010). Novel hysteretic noisy chaotic neural network for broadcast scheduling problems in packet radio networks. *IEEE Transactions on Neural Networks*, 21(9), 1422–1433.

16. Chakrabortya, G., Chakraborty, D., & Shiratori, N. (2005). A heuristic algorithm for optimum transmission schedule in broadcast packet radio networks. *Computer Communications*, 28, 74–85.
17. Gunasekaran, R., Siddharth, S., Krishnaraj, P., Kalaiarasan, M., & Rhymend Uthariaraj, V. (2010). Efficient algorithms to solve broadcast scheduling problem in WiMAX mesh networks. *Computer Communications*, 33, 1325–1333.
18. Ramanathan, S., & Lloyd, E. L. (1993). Scheduling algorithms for multihop radio networks. *IEEE/ACM Transactions on Networking*, 1(2), 166–177.
19. Yeo, J., Lee, H., & Kim, S. (2002). An efficient broadcast scheduling algorithm for TDMA ad-hoc networks. *Computers & Operations Research*, 29(13), 1793–1806.
20. Bi, W., Tang, Z., Wang, J., & Cao, Q. (2005). An improved neural network algorithm for broadcast scheduling problem in packet radio. *Neural Information Processing-Letters and Reviews*, 9(1), 23–29.
21. Menon, S. (2009). A sequential approach for optimal broadcast scheduling in packet radio networks. *IEEE Transactions on Communications*, 57(3), 764–770.
22. Wang, L., & Shi, H. (2006). A gradual noisy chaotic neural network for solving the broadcast schedule problem in packet radio networks. *IEEE Transactions on Neural Networks*, 17(4), 989–1000.
23. Oki, E., & Iwaki, A. (2010). Load-balanced IP routing scheme based on shortest paths in hose model. *IEEE Transactions on Communications*, 58(7), 2088–2096.
24. Wang, L., Liu, W., & Shi, H. (2009). Delay-constrained multicast routing using the noisy chaotic neural networks. *IEEE Transactions on Computers*, 58(1), 82–89.
25. Sinanoglu, O., Karaata, M. H., & Albdaiwi, B. (2010). An inherently stabilizing algorithm for node-to-node routing over all shortest node-disjoint paths in hypercube networks. *IEEE Transactions on Computers*, 59(7), 995–999.
26. Michalewicz, Z. (1995). *Genetic algorithms ? data structure = evolution programs*. New York: Springer.
27. De Jong, K. A., & Spears, W. M. (1989). Using genetic algorithms to solve NP-complete problems. In *Proceedings of the 3rd International Conference on Genetic Algorithms* (pp. 124–132).
28. Banos, R., Gil, C., Reca, J., & Montoya, F. G. (2010). A memetic algorithm applied to the design of water distribution networks. *Applied Soft Computing*, 10, 261–266.
29. Krasnogor, N., & Gustafson, S. (2002). Toward truly memetic algorithms: Discussion and proof of concepts. In *Proceedings of PPSN VII*.

30. Moscato, P. (1999). *Memetic algorithms: A short introduction new ideas in optimization* (pp. 219–234). New York: McGraw Hill.

31. Lu, Z., & Hao, J. K. (2010). A memetic algorithm for graph coloring. *European Journal of Operational Research*, 203, 241–250.

32. Liu, M., Pang, W., Wang, K. P., Song, Y. Z., & Zhou, C. G. (2006).

Improved immune genetic algorithm for solving flow shop scheduling problem. In *Computational methods* (pp. 1057–1062). Springer.

33. Jiao, L., & Wang, L. (2000). A novel genetic algorithm based on immunity. *IEEE Transactions on Systems, Man, and Cybernetics— Part A: Systems And Humans*, 30(5), 552–561.

34. Wang, D., Fung, R. Y. K., & Ip, W. H. (2009). An immunegenetic algorithm for introduction planning of new products.

Computers & Industrial Engineering, 56, 902–917.

Author Biographies

D. Arivudainambi is currently an Associate Professor in the Department of Mathematics,

Anna University, Chennai. He had been at University of Toronto, Toronto. His main research interests include network optimization, network models, queueing models with computer applications and communication systems.

He is reviewer of many journals including IEEE, Elsevier,

Springer and Wiley. He published many papers in reputed journals. He served as General and Technical Program Chair of numerous conferences.

D. Rekha received her B.Sc. and M.Sc. degrees in computer science from Bharathiyar University in 2001 and 2003 respectively.

She is currently working towards the Ph.D. degree in Anna University, Chennai. Her research areas cover wireless multihop networks and evolutionary algorithms