ارائهٔ روشی برای بخش‌بندی و برون‌سپاری اجرای کاربردهای مبتنی بر خدمات وب در سامانه‌های سیار با محدودیت تبادل داده

سیاوش زاهدی[1]، صالح یوسفی[2] و وحید سلوک*[3]
[1و3] دانشکده مهندسی فناوری اطلاعات، دانشگاه صنعتی ارومیه، ارومیه، ایران
[2] دانشکده فنی و مهندسی، دانشگاه ارومیه، ارومیه، ایران

چکیده

برون‌سپاری اجرا به‌عنوان یک راه حل مهم برای اجرای برنامه‌های کاربردی سنگین روی سامانه‌های تلفن همراه است. در سامانه‌های تلفن همراه تغییرات پهنای‌باند در دسترس، به‌طور معمول اتفاق می‌افتد که روی بخش‌بندی بهینه تأثیر می‌گذارد. به‌منظور اجتناب از تکرار این فرایند سنگین بخش‌بندی باید به‌صورت تطبیقی و یک بار انجام شده و با تغییرات پهنای‌باند سازگار باشد. در این مقاله با درنظرگرفتن تغییرات پهنای‌باند و محدودیت تبادل داده، مسألهٔ بخش‌بندی تطبیقی و برون‌سپاری اجرای کاربردهای مبتنی بر خدمات وب به‌صورت سه مدل جداگانه با اهداف متفاوت شامل بهینه‌سازی زمان اجرا، بهینه‌سازی مصرف انرژی و بهینه‌سازی ترکیب وزن‌دار زمان اجرا و مصرف انرژی، فرموله شده و روشی ابتکاری مبتنی بر الگوریتم ژنتیک برای حل هر مسألهٔ بهینه‌سازی در زمان معقول ارائه شده است. نتایج شبیه‌سازی و ارزیابی الگوریتم پیشنهادی نشان می‌دهدکه در مقابل تغییرات پهنای‌باند در دسترس سامانه سیار، عملکرد الگوریتم ارائه‌شده به‌نحو قابل ملاحظه‌ای بهتر از کارهای مشابه است.

واژگان کلیدی: برون‌سپاری اجرا، بخش‌بندی، تطبیق با پهنای‌باند، خدمات وب، محدودیت تبادل داده.

Design and Evaluation of a Method for Partitioning and Offloading Web-based Applications in Mobile Systems with Bandwidth Constraints

Siawash Zahedi[1], Saleh Yousefi[2] & Vahid Solouk[*3]
[1,3]Faculty of IT and Computer Engineering, Urmia University of Technology, Urmia, Iran
[2]Faculty of Engineering, Urmia University, Urmia, Iran

Abstract

Computation offloading is known to be among the effective solutions of running heavy applications on smart mobile devices. However, irregular changes of a mobile data rate have direct impacts on code partitioning when offloading is in progress. It is believed that once a rate-adaptive partitioning performed, the replication of such substantial processes due to bandwidth fluctuation can be avoided. Currently, a wide range of mobile applications are based on web services, which in turn influences the process of offloading and partitioning. As a result, mobile users are prone to face difficulties in data communications due to cost of preferences or connection quality. Taking into account the fluctuations of mobile connection bandwidth and thereby data rate constraints, the current paper proposes a method of adaptive partitioning and computation offloading in three forms. Accordingly, an optimization problem is primarily formulated to each of three main objectives under the investigation. These objectives include run time, energy consumption and the weighted composition of run time and energy consumption. Next, taking into consideration the time complexity of the optimization problems, a heuristic partitioning method based on Genetic Algorithm (GABP) is proposed to solve each of

* Corresponding author
*نویسندهٔ عهده‌دار مکاتبات

● تاریخ ارسال مقاله:۱۳۹۶/۹/۲۴ ● تاریخ پذیرش: ۱۳۹۸/۳/۲۹ ● تاریخ انتشار:۱۳۹۸/۱۰/۰۷ ● نوع مطالعه: پژوهشی

the three objectives and with the capability of acceptable performance maintenance in both dynamic and static partitionings. In order to evaluate and analyze the performance of the proposed approach, a simulation framework was built to run for random graphs of different sizes with the capability of setting specific bandwidth limits as target. The simulation results evidence improved performance against bandwidth fluctuations when compared to similar approaches. Moreover, it was also seen that once the problem circumstances are modified, the offloading can take place in the vicinity of the target node. Furthermore, we implemented the proposed method in form of an application on Android platform to conduct experiments on real applications. The experiments prove that those partitions of the applications requiring higher processing reqources rather than data rate are the best candidates for offloading.

Keywords: Computation offloading, partitioning, bandwidth adaptation, web service, data rate limitation

۱- مقدمه

برنامه‌های کاربردی تلفن همراه به‌طور چشم‌گیری فراگیر شده‌اند و قابلیت‌های غنی‌تر از همیشه را بر روی دستگاه‌های تلفن همراه ارائه می‌کنند. با افزایش پیچیدگی برنامه‌های کاربردی، نیاز آن‌ها به منابع محاسباتی نیز افزایش پیدا می‌کند. پیشرفت در سخت‌افزار و عمر باتری تلفن‌های هوشمند در پاسخ‌گویی به افزایش خواسته‌های محاسباتی برنامه‌های کاربردی آهسته بوده است؛ ازاین‌رو بسیاری از برنامه‌های کاربردی به‌دلیل محدودیت‌های توان محاسباتی و عمر باتری تلفن‌های هوشمند، برای اجرا روی آن‌ها نامناسب هستند. برون‌سپاری اجرا[1] یک ایدهٔ امیدوارکننده برای غلبه بر محدودیت‌های تلفن‌های هوشمند است. برون‌سپاری اجرا به معنای انتقال اجرای برنامه (محاسبات سنگین) به سرورهای قدرتمند و دریافت نتایج از آن‌ها است [3-1].

برای برقرارشدن بهترین مصالحه بین زمان اجرا/مصرف انرژی و هزینهٔ انتقال باید تعیین شود که چه بخش‌هایی از برنامهٔ کاربردی روی تلفن همراه و چه بخش‌هایی روی سرور اجرا شود، که این کار بخش‌بندی نامیده می‌شود [1,4]. به‌طور شهودی می‌توان گفت اجزایی از برنامهٔ کاربردی برای برون‌سپاری مناسبند که نیازمند محاسبات زیاد و هزینهٔ ارتباطی کمی هستند [5]. بخش‌بندی به‌صورت پویا و ایستا انجام می‌گیرد. در روش پویا، در زمان اجرا و با توجه به مقدار فعلی پارامترها (مانند پهنای باند) بخش‌بندی تعیین و درصورت تغییر پارامترها بخش‌بندی دوباره محاسبه می‌شود؛ امّا در بخش‌بندی ایستا پارامترها تخمین زده می‌شوند و بخش‌بندی در زمان اجرا تغییر نمی‌کند [1]. در صورت تغییر پارامترها مانند کاهش پهنای‌باند، روش ایستا عملکرد مناسبی نخواهد داشت؛ از طرف دیگر روش پویا نیز سربار زیادی دارد که شامل اندازه‌گیری پارامترها و محاسبهٔ بخش‌بندی بهینه و تغییر آن در حین اجرای برنامه می‌شود؛ به‌علاوه تغییر بخش‌بندی در حین اجرا ممکن است، برای بسیاری از برنامه‌های کاربردی امکان‌پذیر نباشد؛ بنابراین برای بهبود عملکرد بخش‌بندی ایستا و جلوگیری از سربار بخش‌بندی پویا بخش‌بندی تطبیقی با پهنای‌باند مطرح می‌شود. در این حالت تغییرات پهنای‌باند نیز در نظر گرفته می‌شود و در آغاز اجرای برنامه بخش‌بندی به‌صورتی انتخاب شود که در صورت کاهش پهنای‌باند نیز عملکرد قابل قبولی داشته باشد. در ادامه، این موضوع با یک مثال نشان داده شده است.

(جدول-۱): زمان‌اجرای برنامه‌ی کاربردی نمونه در شکل (۱) در پهنای باندهای kB/sec ۲۰ و kB/sec ۸۰ با بخش‌بندی‌های محاسبه‌شده

(Table-1): Sample Application Runtimes of Fig. 1 using Partitioning

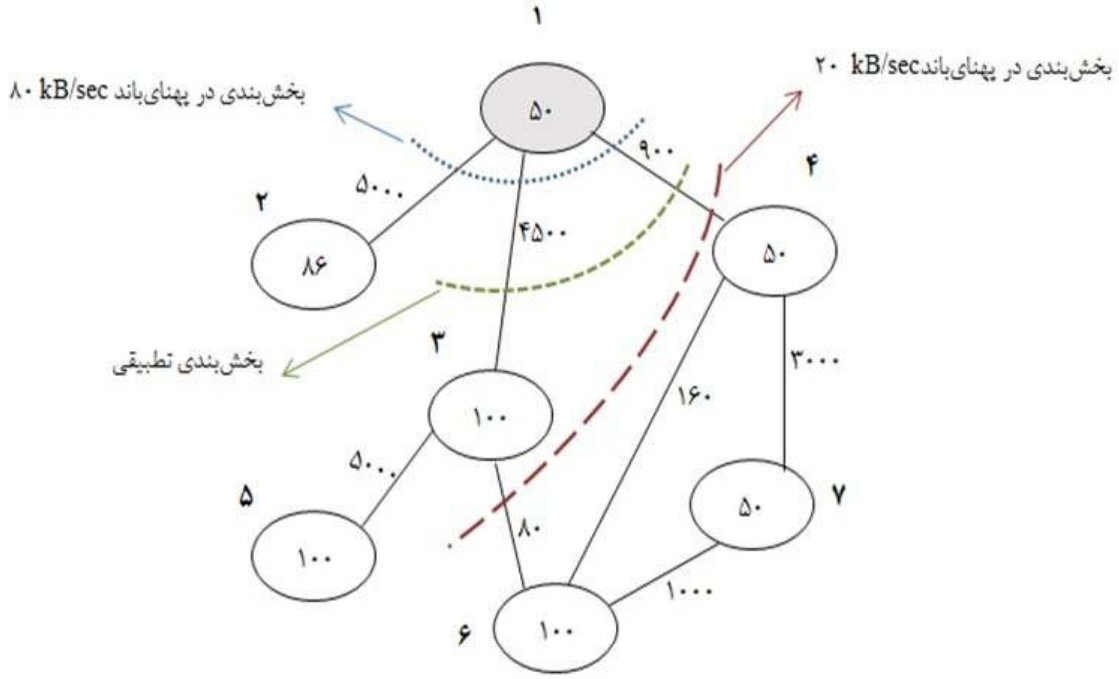
	kB/sec ۲۰	kB/sec ۸۰
بخش‌بندی ایستا در پهنای‌باند ۲۰ kB/sec (برون‌سپاری گره‌های ۴، ۶ و ۷)	۴۲۵	۳۸۸/۲۵
بخش‌بندی ایستا در پهنای‌باند ۸۰ kB/sec (برون‌سپاری گره‌های۲، ۳، ۴، ۵، ۶ و ۷)	۶۶۷/۲	۲۷۷/۲
بخش‌بندی تطبیقی (برون‌سپاری گره‌های ۳، ۴، ۵، ۶ و ۷)	۴۸۶	۲۸۳/۵

در شکل (۱) نحوهٔ نمایش یک برنامهٔ کاربردی نمونه با استفاده از WORGs[2] [4] نشان داده شده است که در آن وزن یال‌ها بیان‌گر هزینهٔ ارتباطی، و وزن رأس‌ها زمان اجرا روی تلفن هوشمند را نشان می‌دهد. با فرض گرهٔ یک به‌عنوان گرهٔ محلی و سرعت سرور پنج برابر سرعت تلفن هوشمند، سه نوع بخش‌بندی محاسبه‌شده، شامل بخش‌بندی تطبیقی و بخش‌بندی در پهنای‌باندهای kB/s ۲۰ و kB/s ۸۰ نمایش داده شده است. در جدول (۱) زمان‌اجرای برنامه‌ی کاربردی نمونه در شکل (۱) در پهنای‌باند kB/s ۲۰ و kB/s ۸۰ توسط بخش‌بندی‌های محاسبه‌شده، نشان داده شده است. جدول (۱) نشان می‌دهد که در صورت تغییر پهنای‌بند، روش ایستا عملکرد مناسبی نخواهد داشت؛ برای نمونه اگر بخش‌بندی

[1] Computation offloading

[2] Weighted object relation graph

بر اساس پهنای‌باند ۸۰ kB/s محاسبه شود و پهنای‌باند به ۲۰ kB/s کاهش پیدا کند، زمان اجرای برنامهٔ کاربردی از (s) ۲۷۷/۲ به (s) ۶۶۷/۲ افزایش پیدا خواهد کرد؛ درحالی‌که زمان اجرا تحت بخش‌بندی ۲۰ kB/s، (s) ۴۳۵ است؛ امّا بخش‌بندی تطبیقی که مانند روش ایستا یک‌بار محاسبه می‌شود و در حین اجرای برنامه کاربردی تغییر نمی‌کند، عملکرد بسیار بهتری نسبت به روش ایستا دارد؛ به‌طوری‌که در پهنای‌باند ۸۰ kB/s عملکرد آن با بخش‌بندی در پهنای‌باند ۸۰ kB/s تنها (s) ۶/۵ اختلاف داشته است؛ امّا در پهنای‌باند ۲۰ kB/s زمان اجرا تحت بخش‌بندی تطبیقی (s) ۴۸۶ بوده درحالی‌که زمان اجرا تحت بخش‌بندی ۸۰ kB/s، (s) ۶۶۷/۲ بوده است. در روش پویا بر اساس پهنای‌باند جدید بخش‌بندی محاسبه می‌شود؛ امّا همان‌طور که پیش‌تر اشاره شد، این روش محدودیت‌هایی دارد.



(شکل-۱): نمایش برنامهٔ کاربردی با استفاده از WORGs
(Figure-1): Demonstration of Application using WORGs

در این مقاله سه مدل بخش‌بندی تطبیقی با تغییرات پهنای‌باند، با اهداف متفاوت شامل بهینه‌سازی ۱) زمان اجرا، ۲) مصرف انرژی و ۳) ترکیب وزن‌دار زمان اجرا و مصرف انرژی به‌منظور بخش‌بندی برنامه‌های کاربردی موبایل ارائه می‌شود. در طراحی مدل‌های پیشنهادی، وجود خدمات وب در برنامهٔ کاربردی و محدودیت‌های احتمالی تبادل داده مورد تأکید ویژه قرار گرفته است. در ادامه با توجه به پیچیدگی زمانی بالای این مدل‌های بهینه‌سازی یک روش بخش‌بندی مبتنی بر الگوریتم ژنتیک با نام GABP پیشنهاد می‌شود که بخش‌بندی مناسبی را در زمان معقول انجام می‌دهد. مدل‌ها و روش پیشنهادی، قابلیت استفاده در بخش‌بندی ایستا و پویا را نیز دارد. نتایج شبیه‌سازی روی گراف‌های تصادفی با اندازه‌های متفاوت نشان می‌دهد که الگوریتم پیشنهادی نسبت به روش‌های دیگر بهبود کارایی قابل توجهی دارد. همچنین نتایج نشان می‌دهد که استفاده از خدمات وب می‌تواند با تغییر شرایط مسأله حتی منجر به برون‌سپاری گره‌های دیگر نیز شود؛ به‌علاوه، یکی دیگر از ویژگی‌های مدل‌ها و الگوریتم پیشنهادی، امکان تعیین اولویت یا ترجیح کاربر در مصالحه وزن‌دار است؛ به‌عنوان نمونه، درصورت انتخاب کاربر برای مصرف پهنای باند کمتر، مصالحه با اختصاص وزن کمتری برای تبادل داده در مقابل زمان اجرای کد و میزان مصرف انرژی صورت می‌گیرد. در انتها نحوهٔ بخش‌بندی و برون‌سپاری یک برنامهٔ کاربردی اندروید ارائه و در یک آزمون تجربی نیز نشان داده می‌شود که اجزایی از برنامهٔ کاربردی که محاسبات زیاد و تبادل دادهٔ کمی نیاز دارند، نامزدهای بهتری برای برون‌سپاری هستند.

ادامهٔ مقاله به‌شکل زیر سازماندهی شده است. در بخش ۲ کارهای پیشین و در بخش ۳ فرضیات مسأله بیان شده است. در بخش ۴ مدل‌های بخش‌بندی پیشنهادی، در بخش ۵ الگوریتم بخش‌بندی و در بخش ۶ ارزیابی کارایی الگوریتم

پیشنهادی با توجه به مدل‌ها ارائه و در بخش ۷ بخش‌بندی و برون‌سپاری یک برنامهٔ کاربردی اندروید توضیح داده شده و درنهایت در بخش ۸ نتیجه‌گیری انجام گرفته است.

۲- کارهای پیشین

پژوهش‌های متعددی در زمینهٔ برون‌سپاری اجرا و بخش‌بندی برنامه‌های کابردی تلفن همراه انجام گرفته که در ادامه به برخی از آن‌ها اشاره شده است. کـارهای بررسی‌شده در این بخش در جدول (۲) آورده شده است.

(جدول-۲): مقایسهٔ برخی کارهای انجام‌شده با رویکرد ایستا و پویا
(Table-2): List of Related works with Static/Dynamic Approaches

مرجع	رویکرد	هدف	پیاده‌سازی
[6]	ایستا	کاهش زمان اجرا و مصرف انرژی	آزمون تجربی(سه برنامهٔ کاربردی اندروید) – پیاده‌سازی روی ماشین مجازی – فرآیند خودکار – offline profiling – ساختن پایگاه‌داده‌ای از بخش‌بندی
[7]	ایستا	کاهش مصرف انرژی با وجود محدودیت توان ارسال و زمان اجرا	شبیه‌سازی روی گراف تصادفی
[8]	ایستا	اجرا با وجود محدودیت زمانی- کاهش مصرف انرژی	آزمون تجربی (سه برنامهٔ کاربردی اندروید) – پیش‌بینی زمان اجرا و پهنای‌باند
[9]	ایستا	مقایسه‌ی مصرف انرژی در حالت اجرای محلی و اجرا روی سرور	آزمون تجربی– پیاده‌سازی الگوریتم مرتب‌سازی (در اندروید) و برون‌سپاری آن روی سرور
[10]	ایستا	کاهش میزان تبادل داده بین مؤلفه‌های برون‌سپاری شده در محیط ابر	اارائه‌ی الگوریتم‌های بخش‌بندی- شبیه‌سازی روی گراف تصادفی
[11]	پویا	کاهش مصرف انرژی	آزمون تجربی(سه برنامه‌ی اندروید) – فرآیند خودکار (با دخالت جزئی توسعه‌دهنده)
[12]	پویا	کاهش زمان اجرا و مصرف انرژی	آزمون تجربی (دو برنامهٔ اندروید) – فرآیند خودکار (استفاده از bytecode)
[13]	نیمه‌پویا	کاهش زمان اجرا	آزمون تجربی(دو برنامهٔ اندروید) –نیاز به پیاده‌سازی برنامه توسط توسعه‌دهنده در دو نسخهٔ سمت کلاینت و سرور
[14]	پویا	کاهش زمان اجرا و مصرف انرژی و استفاده از حافظه	آزمون تجربی (دو برنامهٔ اندروید) –فرآیند خودکار (تنها نیاز به اضافه‌کردن میان افزار به‌صورت فایل کتاب‌خانه‌ای به پروژه در زمان توسعه) –توسعه به‌صورت سرویس
[15]	پویا	کاهش زمان اجرا، مصرف انرژی و تأخیر	فرآیند خودکار
[15]	پویا	کاهش مصرف انرژی در دستگاه‌های IoT	فرآیند خودکار- شبیه‌سازی
[4]	ایستا (بهبود یافته)	کاهش زمان اجرا و مصرف انرژی (مدل وزن‌دار)	ارائهٔ مدل‌ها و الگوریتم‌های بخش‌بندی با در نظر گرفتن تغییرات پهنای‌باند- شبیه‌سازی روی گراف واقعی و تصادفی

در [6] برون‌سپـاری اجـرا در سطح نخ و با هدف کاهش زمان اجرا و مصرف انرژی روی تلفن همراه انجام می‌گیرد (روی سرور، clone ساخته می‌شود). این روش نیازمند تغییر لایهٔ کاربرد مـاشین‌های مجازی مانند Java VM و Dalvik VM اسـت. در ایـن روش تحت شرایـط مختلف بخش‌بندی‌های متفاوتی محـاسبه و در پایگاه‌داده ذخیره و

در هنگام اجرا بر اساس شرایط یکی از بخش‌بندی‌ها روی bytecode برنامه‌ی کاربردی پیاده‌سازی می‌شود. در [7] با درنظرگرفتن یک محدودیت توان ارسال و یک محدودیت تأخیر (با توجه به زمان اجرا و زمان انتقال دادهٔ مورد نیاز)، بخش‌بندی و تخصیص منابع رادیویی (توان ارسال و constellation size) با هم (jointly) به‌نحوی انتخاب می‌شود که مصرف انرژی برنامهٔ کاربردی روی تلفن همراه کمینه شود. در [8] یک معماری برای برون‌سپاری ارائه شده است. کاربر یک زمان تأخیر را مشخص می‌کند، اگر زمان اجرای برنامهٔ کاربردی (زمان اجرا تخمین زده می‌شود) بیشتر از تأخیر تعیین شده باشد، برنامهٔ کاربردی روی تلفن هوشمند و ابر بخش‌بندی می‌شود. در غیر این صورت مصرف انرژی بررسی می‌شود و اگر برون‌سپاری موجب کاهش مصرف انرژی شود برنامهٔ کاربردی بخش‌بندی می‌شود.

در [9] یک مدل مقایسه‌ای برای مصرف انرژی ارائه شده است و مصرف انرژی هنگام اجرای یک کد با محاسبات سنگین روی تلفن هوشمند در مقایسه با مصرف انرژی انتهابه‌انتها درحالتی‌که کد روی سرور اجرا می‌شود، مقایسه و همچنین بهبود عملکرد و مصرف انرژی در دو حالت استفاده از Wi-Fi (۵۴ Mb/s) و 3G (۴۲ Mb/s) مقایسه شده است. در مرجع [10]، یک روش بخش‌بندی برای مؤلفه‌های برون‌سپاری‌شدهٔ یک برنامهٔ کاربردی ارائه و روش پیشنهادی بر روی تعدادی سرور با ظرفیت محاسباتی مشخص در محیط ابر اجرا شده است. هدف اصلی، کاهش نرخ تبادل داده میان مؤلفه‌های برون‌سپاری‌شده و در نتیجه، کمینه‌کردن تأخیر است. برای به‌دست‌آوردن بخش‌بندی الگوریتم‌های مکاشفه‌ای ارائه شده است. چهارچوب MAUI [11] برون‌سپاری در سطح تابع را پشتیبانی می‌کند، توسعه‌دهنده باید توابع یا رده‌هایی را که قابلیت برون‌سپاری دارند، مشخص کند. در MAUI به‌طور پیوسته پهنای‌باند، تأخیر و مصرف انرژی توسط بخش Profiler اندازه‌گیری می‌شوند و بخش‌بندی بهینه با توجه به مصرف انرژی توسط بخش Solver با استفاده از برنامه‌ریزی خطی محاسبه می‌شود. در [12] برون‌سپاری در سطح رده و با هدف بهبود عملکرد و ذخیرهٔ انرژی انجام می‌گیرد. برای کاهش‌دادن سربار تصمیم‌گیری، رده‌های برنامهٔ کاربردی به‌صورت سلسله‌مراتبی خوشه‌بندی می‌شوند به‌طوری‌که رده‌هایی که با یکدیگر ارتباط زیادی دارند (از نظر تعداد فراخوانی) در یک خوشه قرار می‌گیرند و با یکدیگر برون‌سپاری می‌شوند. در [13] توسعه‌دهنده تعیین می‌کند که چه رده‌هایی از برنامه‌کاربردی اندروید برون‌سپاری شوند و این رده‌ها به‌صورت یک خدمات روی سرور پیاده‌سازی می‌شوند. برای برون‌سپاری در زمان اجرا تنها یک شرط بررسی می‌شود و آن هم در دسترس‌بودن منابع سرور است و اگر منابع سرور در دسترس باشند، خدمات سمت سرور فراخوانی می‌شود؛ در غیراین‌صورت نسخهٔ محلی خدمات (کلاس) روی تلفن هوشمند، اجرا می‌شود. میان‌افزار MACS [14] برای بخش‌بندی و برون‌سپاری خودکار برنامه‌های کاربردی اندروید طراحی شده است. تنها کافی است، توسعه‌دهندگان بخش‌هایی را که نیاز به پردازش زیاد دارند، به‌صورت خدمات [16] پیاده‌سازی و فایل‌های کتابخانه‌ای MACS را به پروژهٔ خود اضافه کنند. در این روش بر اساس مدل‌های هزینه و با توجه به زمان اجرا، میزان حافظه و مصرف انرژی (مصرف انرژی صفحهٔ نمایش نیز در نظر گرفته می‌شود) بخش‌بندی بهینه محاسبه می‌شود. در [15] معماری MVR برای برون‌سپاری اجرای برنامه‌های کاربردی با تعامل/محاسبات زیاد در Cloud Edge و استفاده از منابع مجازی (VRs) ارائه شده است. MVR روی دستگاه موبایل دارای پنج مؤلفه و هر VRs درای شش مؤلفه است. در [15] یک معماری یک‌پارچهٔ سه‌لایهٔ جدید از جمله ابر، MEC وIoT ارائه و یک طرح برون‌سپاری انتخابی برای کاهش مصرف انرژی دستگاه‌های IoT و همچنین کاهش سربارهای سیگنالینگ MEC توسعه داده شده است. در [4] بخش‌بندی تطبیقی با پهنای‌باند و یک الگوریتم مبتنی بر Branch and Bound و یک الگوریتم حریصانه به نام MCGAP ارائه و برای به‌دست‌آوردن بخش‌بندی بهینه در یک پهنای باند مشخص از الگوریتم برش کمینهٔ Stoer_Wagner استفاده شده است.

در کارهای مرورشده در جدول (۱) استفاده از خدمات وب در برنامهٔ کاربردی و تعیین مقدار محدودیت تبادل داده در نظر گرفته نشده است. در مدل‌ها و الگوریتم پیشنهادی این مقاله، علاوه‌بر درنظرگرفتن این موارد در فرآیند مدل‌سازی و پیشنهاد الگوریتم جدید، کارایی به مراتب بهتری از الگوریتم‌های پیشین به‌دست آمده است.

۳- فرضیات

در این بخش فرضیات مسأله بیان می‌شود. در مدل‌های ارائه‌شده در این مقاله فرض شده که پهنای‌باند ارسال و دریافت برابر است و دستگاه تلفن همراه از Wi-Fi به‌عنوان واسط شبکه استفاده می‌کند؛ همچنین توان ارسال و دریافت Wi-Fi نیز برابر فرض شده و از زمان بیکاربودن واسط شبکه برای دریافت پاسخ بخش برون‌سپاری‌شده نیز صرف‌نظر و

همچنین فرض شده است که گراف برنامهٔ کاربردی موجود است. در این مقاله برای بازنمایش برنامهٔ کاربردی از گراف وزن‌دار ارتباط شیء (WORGs) استفاده شده است. WORG، یک گراف غیر جهت‌دار است که در آن رأس‌های رده‌های برنامهٔ کاربردی و یال‌ها ارتباط بین رده‌ها را نشان می‌دهد و وزن رأس‌ها زمان اجرای رده‌ها روی دستگاه تلفن همراه و وزن یال‌ها میزان دادهٔ مبادله‌شده بین آن‌ها را مشخص می‌کنند.

با درنظرگرفتن استفاده از خدمات وب، گره‌های یک برنامهٔ کاربردی به سه دسته تقسیم می‌شوند:

۱. گره‌هایی که باید به‌صورت محلی و روی تلفن همراه اجرا شوند.

۲. گره‌هایی که به‌صورت خدمات وب تعریف می‌شوند (روی تلفن همراه اجرا نمی‌شوند) و شامل دو نوع هستند:

- خدمات وب نوع نخست، بخش‌هایی از برنامهٔ کاربردی که در قالب خدمات وب پیاده‌سازی می‌شوند و زمان اجرای آن‌ها در سامانه محلی (موبایل) معلوم است.
- خدمات وب نوع دوم، وب‌سرویس‌های سازمان‌های دیگر که مورد استفاده قرار می‌گیرند و زمان اجرای آن‌ها به صورت یک مقدار متوسط تعیین می‌شود.

۳. گره‌هایی که قابلیت برون‌سپاری دارند.

۱-۳- مدل‌های پیشنهادی برای بخش‌بندی

در این بخش مدل‌های بخش‌بندی را با توجه به زمان اجرا و مصرف انرژی و با درنظرگرفتن استفاده از خدمات وب در برنامهٔ کاربردی و محدودیت تبادل داده ارائه می‌دهیم. مدل‌های پیشنهادی، گسترش‌یافتهٔ مدل‌های ارائه‌شده در [4] هستند. علائم به‌کاربرده‌شده در مدل‌ها در جدول (۳) آورده شده است. از آن‌جایی‌که برون‌سپاری اجرا با تبادل داده همراه است با تعیین یک محدودیت مبادلهٔ داده $Data_{max}$ می‌توان میزان تبادل داده بین تلفن همراه و سرور را کنترل کرد. در مدل‌های ارائه‌شده، مقدار تبادل داده توسط بخش‌بندی X، یعنی $Data(X)$ طبق رابطهٔ زیر محاسبه می‌شود:

$$Data(X)=\sum_{1\le i<j\le n}\left|x_i-x_j\right|\times W_{ij} \qquad (۱)$$

که در آن W_{ij} دادهٔ مبادله‌شده بین گره‌های i و j و x_i نتیجهٔ بخش‌بندی گره i است ($x_i=1$: گره i به‌صورت محلی روی تلفن همراه اجرا می‌شود؛ و $x_i=0$: گره i روی سرور اجرا می‌شود). رابطه (۱) میزان تبادل داده بین گره‌هایی را نشان می‌دهد که با یکدیگر در ارتباط هستند ($W_{ij}\neq 0$) و در بخش‌بندی X روی یک سامانه اجرا نمی‌شوند ($\left|x_i-x_j\right|=1$).

در ادامه سه مدل برای بخش‌بندی برنامه کاربردی‌های تلفن همراه با در نظر تغییرات پهنای‌باند، استفاده از خدمات وب در برنامه کاربردی و محدودیت مبادلهٔ داده ارائه می‌شود.

۱-۱-۳- مدل بهینه‌سازی زمان اجرا

هدف انتخاب یک بخش‌بندی X است، به‌طوری‌که:

- زمان اجرای برنامهٔ کاربردی توسط بخش‌بندی X تحت پهنای‌باند b، یعنی $T_{ws}(X,b)$ کمینه باشد؛
- زمان اجرای برنامهٔ کاربردی توسط بخش‌بندی X در پهنای‌باند b_l، یعنی $T_{ws}(X,b_l)$ از یک حدی بیشتر نباشد؛
- میزان تبادل داده میان تلفن همراه و سرور توسط بخش‌بندی X یعنی، $Data(X)$ بیشتر از یک حد تعیین‌شده، یعنی $Data_{max}$ نباشد.
- به این ترتیب فضای راه‌حل مدل Xd برای محاسبهٔ بخش‌بندی تطبیقی با پهنای‌باند با درنظرگرفتن محدودیت تبادل داده، این گونه تعریف می‌شود:

$$X_d=\left\{X\ \middle|\ T_{ws}(X,b_l)\le T_{ws}\left(X_{b_l}^{opt},b_l\right)\times(1+\alpha)\ and\ Data(X)\le Data_{max}\right\} \qquad (۲)$$

- که در آن $T_{ws}\left(X_{b_l}^{opt},b_l\right)$ زمان اجرای برنامهٔ کاربردی توسط بخش‌بندی بهینهٔ $X_{b_l}^{opt}$ در پهنای‌باند b_l است. ثابت تجربی a $(a>0)$، زمان اجرای $T_{ws}(X,b_l)$ را به حد بالای $T_{ws}\left(X_{b_l}^{opt},b_l\right)\times(1+\alpha)$ محدود می‌کند [4].

اگر مقدار a خیلی کوچک انتخاب شود، ممکن است بخش‌بندی که شرط نخست موجود در X_d را برآورد کند، وجود نداشته باشد؛ اگر مقدار a بزرگ انتخاب شود، به‌طوری‌که $T_{ws}\left(X_b^{opt},b_l\right)/T_{ws}\left(X_{b_l}^{opt},b_l\right)\le(1+\alpha)$ در این صورت بخش‌بندی بهینه تحت پهنای باند b، یعنی $X_{b_l}^{opt}$ به‌عنوان راه حل انتخاب می‌شود.

زمان اجرای برنامهٔ کابردی $T_{ws}(X,b)$ توسط بخش‌بندی X تحت پهنای‌باند b طبق رابطهٔ زیر تعریف می‌شود:

$$T_{WS}(X,b)=\sum_{1\le i\le n}\left(x_i\times t_{nli}+(1-x_i)\times t_{nsi}\right)+\sum_{1\le i<j\le n}\left|x_i-x_j\right|\times t_{ij}+\sum_{1\le i\le n_y}t_{wsi} \qquad (۳)$$

که در آن زمان اجرای گره i روی سرور (t_{nsi}) برابر است با $t_{nsi}=t_{nli}/k$، با وجود دادهٔ مبادله‌شده W_{ij} و پهنای‌باند b زمان انتقال داده بین گره i و j برابر است با $t_{ij}=W_{ij}/b$، t_{wsi} زمان نهایی اجرای خدمات وب شماره i را نشان می‌دهد و با توجه به نوع خدمات وب محاسبه می‌شود. برای خدمات وب نوع نخست $T_{wsi}=t_{si}/k$ که در آن t_{si} نشان‌دهندهٔ زمان اجرا

روی موبایل است، برای خدمات وب نوع دوم $T_{wsi} = t_{si}$ که در آن t_{si} نشان‌دهندهٔ متوسط زمان اجرای خدمات وب و مستقل از k است. در رابطه (۳) دورهٔ نخست زمان اجرای گره‌ها و دورهٔ دوم هزینهٔ ارتباطی (با توجه به زمان تبادل داده) بین گره‌هایی را نشان می‌دهد که با یک دیگر در ارتباط هستند (در گراف برنامهٔ کاربردی بین آن‌ها یال وجود دارد) و روی یک سامانه اجرا نمی‌شوند و دورهٔ سوم زمان اجرای خدمات وب را نشان می‌دهد که برای گره‌های خدمات وب، t_{nli} و x_i صفر است. برای محاسبهٔ زمان اجرای برنامهٔ کاربردی به‌صورت محلی $T_{LocalWS}$ *(s)* باید در رابطه (۳) برای گره‌های خدمات وب $x_i=0$ و بقیهٔ گره‌ها $x_i=1$ قرار داده شود، وجود خدمات وب در برنامهٔ کاربردی موجب می‌شود که در حالت اجرای محلی نیز هزینهٔ ارتباطی وجود داشته باشد؛ به این ترتیب تغییرات پهنای‌باند روی زمان اجرای برنامهٔ کاربردی به‌صورت محلی تأثیر می‌گذارد.

رابطهٔ (۲) یک مسألهٔ بهینه‌سازی است که برای به‌دست‌آرودن بخش‌بندی (تطبیقی) بهینه در بدترین حالت، لازم است مقایسه بین 2^n بخش‌بندی ممکن صورت پذیرد. در این مقاله در بخش ۵ روشی مبتنی بر الگوریتم ژنتیک برای آن ارائه می‌کنیم.

(جدول-۳): علائم به‌کار رفته در مدل بخش‌بندی پیشنهادی
(Table-3): Notations for the proposed partitioning method

برچسپ	پارامتر
$Data(X)$ *(KB)*	مقدار دادهٔ مبادله‌شده بین تلفن همراه و سرور تحت بخش‌بندی X
$Data_{max}$ *(KB)*	بیشینهٔ دادهٔ مبادله‌شدهٔ مجاز بین تلفن همراه و سرور
$T_{ws}(X,b)$ *(s)*	زمان اجرای برنامه کاربردی توسط بخش‌بندی X تحت پهنای‌باند b
X	بخش‌بندی برنامه کاربردی که به صورت $X = x_1,x_2, \dots, x_n$ تعریف می‌شود
x_i	نتیجهٔ بخش‌بندی گره i، $x_i=1$: نود i به‌صورت محلی روی تلفن همراه اجرا می‌شود؛ و $x_i=0$: گره i روی سرور اجرا می‌شود، $i= 1...n$
N	تعداد کل گره‌های برنامهٔ کاربردی
n_s	تعداد گره‌های خدمات وب
b *(kB/s)*	پهنای‌باند فعلی
b_l *(kB/s)*	پهنای‌باند کمینه
t_{nli} *(s)*	زمان اجرای محلی گرهٔ i روی تلفن همراه
t_{nsi} *(s)*	زمان اجرای گرهٔ i روی سرور
t_{si} *(s)*	زمان اجرای خدمات وب i را نشان می‌دهد
t_{wsi} *(s)*	زمان نهایی اجرای خدمات وب I را نشان می دهد
K	نسبت زمان اجرای یک گره روی سرور به زمان اجرای آن روی تلفن همراه
W_{ij} *(KB)*	دادهٔ مبادله‌شده بین گره‌های i و j
t_{ij} *(s)*	زمان انتقال داده بین گره‌های i و j
$E_{ws}(X,b)$ *(J)*	مصرف انرژی برنامه کاربردی توسط بخش‌بندی X تحت پهنای‌باند b
E_i *(J)*	انرژی مصرفی گرهٔ i زمانی که به‌طور محلی روی موبایل اجرا شود
E_{ij} *(J)*	انرژی مصرفی برای انتقال داده بین گره‌های i و j
P_{cpu} *(W)*	توان CPU موبایل
P_w *(mW)*	توان ارسال و دریافت واسط شبکه (Wi-Fi)
$T_{LocalWS}$ *(s)*	زمان اجرا در حالتی که کل برنامهٔ کاربردی به‌صورت محلی اجرا می‌شود.
$E_{LocalWS}$ *(J)*	انرژی مصرفی در حالتی که کل برنامهٔ کاربردی به‌صورت محلی اجرا می‌شود.
$WTE_{ws}(X,b)$	مقدار عددی وزن‌دار (زمان اجرا و مصرف انرژی) توسط بخش‌بندی X و تحت پهنای‌باند b
w_t	وزن زمان اجرا در مدل بخش‌بندی وزن‌دار
w_e	وزن انرژی مصرفی در مدل بخش‌بندی وزن‌دار به طوری که: $w_t + w_e = 1$

۲-۱-۳- مدل بهینه‌سازی مصرف انرژی

هدف انتخاب یک بخش‌بندی X است به‌طوری‌که:

- مصرف انرژی برنامهٔ کاربردی توسط بخش‌بندی X تحت پهنای‌باند b، یعنی $E_{ws}(X,b)$ کمینه باشد؛

- مصرف انرژی برنامهٔ کاربردی توسط بخش‌بندی X در پهنای‌باند b_l، یعنی $E_{ws}(X,b_l)$ از یک حدی بیشتر نباشد؛
- میزان تبادل داده میان تلفن همراه و سرور توسط بخش‌بندی X یعنی، $Data(X)$ بیشتر از یک حد تعیین‌شده، یعنی $Data_{max}$ نباشد.

فضای راه‌حل مدل X_d برای محاسبهٔ بخش‌بندی تطبیقی با پهنای‌باند با درنظرگرفتن محدودیت تبادل داده، این گونه تعریف می‌شود:

$$X_d=\left\{X \mid E_{ws}(X,b_l)\le E_{ws}\left(X_{b_l}^{opt},b_l\right)\times(1+\alpha)\ and\ Data(X)\le Data_{max}\right\} \quad (۴)$$

که در آن $E_{ws}\left(X_{b_l}^{opt},b_l\right)$ مصرف انرژی برنامهٔ کاربردی توسط بخش‌بندی بهینهٔ $X_{b_l}^{opt}$ در پهنای‌باند b_l است. ثابت تجربی a $(a>0)$، مصرف انرژی $E_{ws}(X,b_l)$ را به حد بالای $E_{ws}\left(X_{b_l}^{opt},b_l\right)\times(1+\alpha)$ محدود می‌کند [4]. اگر مقدار a خیلی کوچک انتخاب شود، ممکن است بخش‌بندی که شرط نخست موجود در X_dرا براورد کند، وجود نداشته باشد؛ اگر مقدار a بزرگ انتخاب شود، به‌طوری‌که $E_{ws}\left(X_b^{opt},b_l\right)/E_{ws}\left(X_{b_l}^{opt},b_l\right)\le(1+a)$ در این صورت بخش‌بندی بهینه تحت پهنای باند b، یعنی $X_{b_l}^{opt}$ به‌عنوان راه حل انتخاب می‌شود.

مصرف انرژی برنامهٔ کابرد ی $E_{ws}(X,b)$ توسط بخش‌بندی X تحت پهنای‌باند b طبق رابطهٔ زیر تعریف می‌شود:

$$E_{ws}(X,b)=\sum_{1\le i\le n}(x_i\times E_i)+\sum_{1\le i<j\le n}\left|x_i-x_j\right|\times E_{ij} \quad (۵)$$

که در آن $E_i=t_{nli}\times P_{cpu}$ و $E_{ij}=t_{ij}\times P_w$. در رابطه (۵) دورهٔ نخست انرژی مصرفی گره‌ها و دورهٔ دوم هزینهٔ ارتباطی (با توجه به انرژی مصرفی برای تبادل داده) بین گره‌هایی را نشان می‌دهد که با یک‌دیگر در ارتباط هستند و روی یک سامانه اجرا نمی‌شوند. برای گره‌های خدمات وب E_i (t_{nli}) و x_i صفر است. برای محاسبهٔ انرژی مصرفی برنامه کاربردی به‌صورت محلی $(E_{LocalWS}(J))$ باید در رابطه (۵) برای گره‌های خدمات وب $x_i=0$ و بقیهٔ گره‌ها $x_i=1$ قرار داده شود. وجود خدمات وب در برنامهٔ کاربردی موجب می‌شود که در حالت اجرای محلی نیز هزینهٔ ارتباطی وجود داشته باشد، به این ترتیب تغییرات پهنای‌باند روی انرژی مصرفی برنامه کاربردی به‌صورت محلی تأثیر می‌گذارد. رابطه (۵) به‌طورکامل مشابه مدل ارائه‌شده برای مصرف انرژی (بدون درنظرگرفتن خدمات وب) در [4] است.

رابطه (۴) یک مسألهٔ بهینه‌سازی است که برای به‌دست‌آوردن بخش‌بندی (بخش‌بندی تطبیقی) بهینه در بدترین حالت بایستی کل 2^n بخش‌بندی ممکن مقایسه شود. در این مقاله در بخش ۵ روشی مبتنی بر الگوریتم ژنتیک برای آن ارائه می‌کنیم.

۳-۱-۳- مدل وزن‌دار بهینه‌سازی زمان اجرا و مصرف انرژی

هدف انتخاب یک بخش‌بندی X است، به‌طوری‌که:

- مقدار عددی وزن‌دار (زمان اجرا و مصرف انرژی) توسط بخش‌بندی X تحت پهنای‌باند b، یعنی $WTE_{ws}(X,b)$ کمینه باشد؛
- مقدار عددی وزن‌دار (زمان اجرا و مصرف انرژی) توسط بخش‌بندی X در پهنای‌باند b_l، یعنی $WTE_{ws}(X,b)$ از یک حدی بیشتر نباشد؛
- میزان تبادل داده میان موبایل و سرور توسط بخش‌بندی X یعنی، $Data(X)$ بیشتر از یک حد تعیین‌شده، یعنی $Data_{max}$ نباشد.

فضای راه‌حل مدل X_d برای محاسبهٔ بخش‌بندی تطبیقی با پهنای‌باند با درنظرگرفتن محدودیت تبادل داده، این گونه تعریف می‌شود:

$$X_d=\left\{X \mid WTE_{ws}(X,b_l)\le WTE_{ws}\left(X_{b_l}^{opt},b_l\right)\times(1+\alpha)\ and\ Data(X)\le Data_{max}\right\} \quad (۶)$$

که در آن $WTE_{ws}\left(X_{b_l}^{opt},b_l\right)$ مقدار مدل وزن‌دار توسط بخش‌بندی بهینهٔ $X_{b_l}^{opt}$ در پهنای‌باند b_l است. ثابت تجربی a $(a>0)$، مقدار مدل وزن‌دار $WTE_{ws}(X,b_l)$ را به حد بالای $WTE_{ws}\left(X_{b_l}^{opt},b_l\right)\times(1+\alpha)$ محدود می‌کند [4]. اگر مقدار a خیلی کوچک انتخاب شود، ممکن است بخش‌بندی که شرط نخست موجود در X_dرا برآورد کند، وجود نداشته باشد؛ اگر مقدار a بزرگ انتخاب شود، به‌طوری‌که $WTE_{ws}\left(X_b^{opt},b_l\right)/WTE_{ws}\left(X_{b_l}^{opt},b_l\right)\le(1+\alpha)$ در این صورت بخش‌بندی بهینه تحت پهنای باند b، یعنی $X_{b_l}^{opt}$ به‌عنوان راه‌حل انتخاب می‌شود.

مدل وزن‌دار بهینه‌سازی زمان اجرا و مصرف انرژی $WTE_{ws}(X,b)$ طبق رابطهٔ زیر تعریف می‌شود:

$$WTE_{ws}(X,b)=w_t\times T_{ws}(X,b)/T_{LocalWS}+w_e\times E_{ws}(X,b)/E_{LocalWS} \quad (۷)$$

که در آن $T_{LocalWS}$ و $E_{LocalWS}$ به‌ترتیب برابر هستند با زمان اجرا و مصرف انرژی زمانی که برنامهٔ کاربردی به صورت محلی روی تلفن همراه اجرا می‌شود. اوزان زمان اجرا (w_t) و مصرف انرژی (w_e) بر اساس اولویت کار به کاهش زمان یا مصرف انرژی تعیین می‌شوند.

رابطه (۶) یک مسألهٔ بهینه‌سازی است که برای به‌دست‌آرودن بخش‌بندی (بخش‌بندی تطبیقی) بهینه در بدترین حالت بایستی کل 2^n بخش‌بندی ممکن مقایسه شود. در این مقاله در بخش ۵ روشی مبتنی بر الگوریتم ژنتیک برای آن ارائه می‌کنیم.

۴- الگوریتم بخش‌بندی پیشنهادی

همان‌طور که پیش‌تر اشاره شد، در یک برنامهٔ کاربردی با n مؤلفه، پیداکردن بخش‌بندی بهینه با در نظرگفتن محدودیت (تغییرات پهنای‌باند و محدودیت تبادل داده) در بدترین حالت مستلزم جست و جو در 2^n حالت از فضای مسأله است که برای برنامه‌هایی با تعداد مؤلفهٔ بزرگ‌تر از بیست، پیداکردن بخش‌بندی، بسیار زمان‌بر و درنتیجه غیرکاربردی خواهد بود؛ بنابراین نیاز به ارائهٔ روش‌هایی است که بتوان یک بخش‌بندی مناسب را در زمان قابل‌قبول محاسبه کرد. در ادامه روش ابتکاری مبتنی بر الگوریتم ژنتیک (GBAP) برای تعیین بخش‌بندی با توجه به مدل‌های پیشنهادی ارائه شده است.

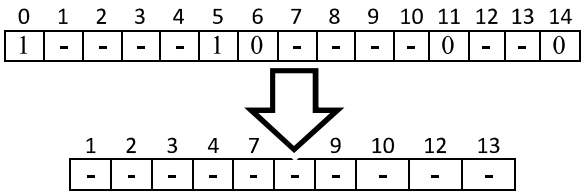
الگوریتم پیشنهادی برای هر سه مدل بخش چهار قابل اعمال است؛ امّا به‌منظور رعایت اختصار در توضیحات شبه‌کد این بخش، از مدل وزن‌دار استفاده می‌شود که حالت کلی بوده و دو مدل قبلی را نیز شامل می‌شود. در ادامه منظور از $WTE_{ws}(X,b)$ مقدار عددی مدل وزن‌دار (زمان اجرا و مصرف انرژی) مطابق رابطه (۷) در بخش ۴-۱-۳- است.

۴-۱- روش ابتکاری مبتنی بر الگوریتم ژنتیک

در این بخش الگوریتم بخش‌بندی پیشنهادی توضیح داده شده است. در روش پیشنهادی از الگوریتم ژنتیک گسسته استفاده شده است. ساختار کروموزوم در شکل (۲) نشان داده شده است. هر ژن نمایندهٔ یک گره است که دارای مقدار یک یا صفر است و طول هر کروموزوم برابر با تعداد گره‌هایی است که وضعیت آن‌ها مشخص نیست، به این ترتیب گره‌هایی که در وضعیت برون‌سپاری قرار دارند و گره‌هایی که باید روی موبایل اجرا شوند، جزو کروموزم نیستند. در روش پیشنهادی از دو استدلال زیر استفاده شده است [4]:

۱. مجموعهٔ گره‌هایی که در بخش‌بندی بهینه تحت پهنای‌باند کمینه (b_l) در وضعیت برون‌سپاری ($x_i=0$) قرار دارند، یعنی S_{b_l} در بخش‌بندی بهینه تحت پهنای‌باندهای بالاتر نیز در این وضعیت خواهند ماند. برای نمونه در بخش یک جدول (۱)، گره‌های ۴، ۵ و ۶ که در بخش‌بندی بهینه تحت پهنای‌باند ۲۰ kB/s برون‌سپاری شده‌اند در بخش‌بندی بهینه تحت پهنای‌باند ۸۰ kB/s نیز برون‌سپاری شده‌اند.

۲. مجموعهٔ گره‌هایی که در بخش‌بندی بهینه تحت پهنای‌باند فعلی (b) در وضعیت اجرای محلی ($x_i=1$) قرار دارند، یعنی C_b در بخش‌بندی بهینه تحت پهنای‌باندهای پایین‌تر نیز در این وضعیت خواهند ماند؛ به این ترتیب طول کروموزوم برابر خواهد بود با $l_{crm} = n - n_{C_b} - n_{s_{bl}}$ که n_{C_b} و $n_{s_{bl}}$ تعداد گره‌های C_b و S_{b_l} هستند. با مشخص‌شدن گره‌های تأثیرگذار، احتمال پیداکردن بخش‌بندی تطبیقی بهینه افزایش می‌یابد. در شکل (۲) ساختار کروموزم بعد از حذف ژن‌هایی (گره‌هایی) که وضعیت آن‌ها مشخص است (گره‌های C_b و S_{b_l})، نشان داده شده است.



(شکل-۲): ساختار کروموزوم

(Figure-2): Chromosome Structure

شبه‌کد روش پیشنهادی در الگوریتم (۱) نشان داده شده است. در شبه‌کد ارائه‌شده *WORG*، گراف برنامهٔ کاربردی است.

در خطوط ۱و۲ برای تعیین حد بالا، مقدار بهینهٔ مدل در پهنای‌باند کمینه b_l محاسبه می‌شود. در خطوط ۳ و ۵، S_{b_l} و C_b بر اساس بخش‌بندی‌های بهینه تحت پهنای باند b و b_l محاسبه و گره‌های خدمات وب N_s (در صورت وجود) به S_{b_l} اضافه می‌شوند؛ زیرا گره‌های خدمات وب به‌حتم باید برون‌سپاری شوند. در خط ۶ و ۷ به‌صورت تصادفی و به‌اندازهٔ جمعیت ($pop_size = 2 \times n$) کروموزوم تولید می‌شود و برای جلوگیری از تولیدشدن کروموزوم‌های تکراری، تعداد $log_2^{pop_size}$ ژن اول کروموزوم‌ها به ترتیب با اعداد دودویی صفر تا pop_size مشخصی و سپس با ترکیب‌کردن (S_{b_l}) و (C_b) با هر کروموزم، بخش‌بندی‌های تصادفی تولید می‌شود (کنترل می‌شود که تا حد امکان تعداد

$pop_size \times mu_rate$ بخش‌بندی معتبر تولید شود) و مقدار مدل تحت بخش‌بندی‌هایی که شرط حد بالا و محدودیت مبادلهٔ داده را نقض می‌کنند در عدد M ضرب می‌شود ($M = 10$) تا اطمینان حاصل شود که این بخش‌بندی‌ها پایین‌تر از بخش‌بندی‌هایی معتبر قرار نمی‌گیرند؛ به‌دلیل اینکه ممکن است، تعداد بخش‌بندی‌های معتبر به‌اندازهٔ $pop_size \times co_rate$ نباشد، بخش‌بندی‌های نامعتبر حذف نمی‌شوند. در خط هشت بخش‌بندی‌ها بر اساس مقدار مدل مرتب می‌شوند (با استفاده از مرتب‌سازی درجی) و مراحل زیر (خطوط ۱۰ تا ۱۴) تا هم‌گراشدن الگوریتم ($pop_size \times co_rate$ بخش‌بندی یکسان) یا رسیدن به بیشینهٔ تعداد نسل (***max_gen***) تکرار می‌شود. در خط‌های ۱۰ و ۱۱ با توجه به نرخ crossover (***co_rate***) و نرخ mutation (***mu_rate***) و با استفاده از عمل‌گرهای mutation و crossover کروموزوم‌های جدید (نسل جدید) تولید می‌شود. انتخاب کروموزوم‌ها برای عمل crossover توسط چرخ رولت و براساس رتبه انجام می‌گیرد و برای تولید $pop_size \times (1 - co_rate)$ کروموزوم جدید از روش two point crossover استفاده می‌شود، به این صورت که دو کروموزوم به‌طور تصادفی انتخاب و در هر دو کروموزوم نیز دو نقطه به‌طور تصادفی انتخاب و ژن‌های بین آنها جابه‌جا می‌شود؛ سپس تعداد $pop_size \times mu_rate$ کروموزوم به‌طور تصادفی از بین کل جمعیت انتخاب و از هرکدام یک گره به‌طور تصادفی انتخاب می‌شود و وضعیت آن تغییر می‌یابد (از صفر به یک یا از یک به صفر)، اگر کروموزوم انتخاب‌شده جز و $pop_size \times mu_rate$ کروموزوم برتر باشد، تغییر وضعیت هنگامی اعمال می‌شود که راه‌حل بهبود بیابد. در خط دوازده مقدار مدل تحت بخش‌بندی‌های جدید تولید شده که شرط حد بالا و محدودیت مبادلهٔ داده را نقض می‌کنند، در عدد M ضرب می‌شود ($M = 10$)، بدیهی است اگر مقدار $Data_{max}$ عددی بزرگ باشد، محدودیت مبادلهٔ داده در نظر گرفته نمی‌شود و در انتخاب بخش‌بندی تطبیقی تأثیری نمی‌گذارد. در ادامه بخش‌بندی‌ها بر اساس مقدار مدل مرتب می‌شوند و به تعداد نسل‌ها یک واحد اضافه می‌شود (خط چهارده). در خط پانزده بهترین راه‌حل X_{elit} که در نسل آخر در مکان اول قرار دارد، به‌عنوان پاسخ انتخاب می‌شود. مرتبهٔ زمانی روش پیشنهادی برابر است با:

$$F(n) = O(n_{pop_size} \times max_gen \times (Fs(n) + Fc(n))) \quad (۹)$$

که در آن $Fc(n) = O(n_{pop_size} \times (l_{crom})^2)$ و $Fs(n)$ به‌ترتیب مرتبهٔ اجرایی مرتب‌سازی و عملگر crossover هستند (l_{crom} طول کروموزوم است).

Input: $WORG, b, b_1, a, pop_size, co_rate, mu_rate, \max_gen, N_s, Data_{max}$
Output: *The best partitioning solution* X_{elit} *and model value* $\left(WTE_{ws}(X_{elit}\ b)\right)$;
Begin:
1: ***Compute*** *the* $WTE_{ws}(X_{elit}\ b_l)$; // $X_{b_l}^{opt}$ *is the optimal partitioning in minimum bandwidth (i.e.* b_l*)*
2: $minValue = WTE_{ws}\left(X_{b_l}^{opt}\ b_l\right)$; *//upper bound, equation (7) in section 4-1-3*
3: ***Add*** *nodes in* $X_{b_l}^{opt}$ *where* $x_l = 0$ *and nodes in* N_s *to* S_{b_l};
4: ***Compute*** $WTE_{ws}\left(X_{b_l}^{opt}\ b_l\right)$
5: ***Add*** *nodes in* $X_{b_l}^{opt}$ *where* $x_i = l$ *and nodes in* C_b; //$X_{b_l}^{opt}$ *is the optimal partitioning in current bandwidth (i.e. b)*
6: ***Initialize*** pop_size;
7: *Exclude non_satisfying solutions; //upper bound and* $Data_{max}$ *are used here*
8: ***Sort*** *the population based on model value;*
9: ***While*** $\left(current_generation \leq \max_{gen} and\ no_convergence()\ \right)$
10: ***Keep*** *first* $pop_size \times co_rate$ *solutions and generate* $pop_size \times (1 - co_rate)$ *solution using crossover;*
11: ***Mutate*** $pop_size \times mu_rate$ *solutions;*
12: ***Exclude*** *non_satisfying solutions; //upper bound and* $Data_{max}$ *are used here*
13: ***Sort*** *the population based on model value;*
14: $(current_generation = current_generation + 1)$;
End *while*
15: ***Return*** X_{elit}, $WTE_{ws}(X_{elit}\ b)$; //X_{elit} *is the best solution (the chromosome located in the first place of the last generation*

(الگوریتم-۱): روش ابتکاری مبتنی بر الگوریتم ژنتیک (GBAP)
(Algorithm-1): GBAP Algoritm

۵- ارزیابی کارآیی

در این بخش کارایی مدل‌ها و الگوریتم بخش‌بندی پیشنهادی (GABP) درصورت وجود یا عدم وجود خدمات وب در برنامه‌ی کاربردی و با درنظرگرفتن تغییرات پهنای‌باند و محدودیت تبادل داده ارزیابی شده است.

در ارزیابی انجام‌شده شش گراف (WORGs) به‌صورت تصادفی تولید شده است (جدول (۴)). پارامترهای مسأله در جدول (۵) نشـان داده شده است. اجرای کوتاه الگوریتم‌ها برای به دست‌آوردن بخش‌بندی روی نوت‌بوک VAIO FZ290 با حافظهٔ GB ۲، CPU با فرکانس GHs ۲ و سامانه‌عامل ویندوز هفت ۳۲ بیتی و اجراهای طولانی روی سرور HP DL380p Gen8 با بهره‌گیری از یک CPU با فرکانس GHs ۲/۶ انجام شده است. در ادامهٔ این مقاله برای به‌دست‌آوردن بخش‌بندی بهینه در یک پهنای‌باند مشخص از الگوریتم برش‌کمینه Stoer–Wagner [4] استفاده شده است.

روش پیشنهادی (GABP) با حالت‌ها و روش‌های زیر مقایسه شده است:

۱. SP: در این حالت بخش‌بندی بهینه یک‌بار و در پهنای‌باند ۱۰۰kB/sec محاسبه شده است (روش ایستا).
۲. DP: در این حالت بخش‌بندی بهینه به دفعات و در هر ۱۰ kB/sec تغییر در پهنای‌باند محاسبه شده است (روش پویا).
۳. MCGA: در این روش، بخش‌بندی تطبیقی با پهنای باند به روش مرجع [4] پیاده‌سازی شده است.
۴. OP: در این حالت بخش‌بندی تطبیقی پهنای‌باند بهینه با جست و جوی کامل فضای حالت مسأله محاسبه شده است.
۵. Local execution: اجرای محلی برنامهٔ کاربردی روی موبایل بدون بخش‌بندی، در این حالت گره‌هایی را از برنامهٔ کاربردی که قابلیت برون‌سپاری دارند، به‌صورت محلی روی تلفن همراه اجرا می‌شوند.

در ارزیابی انجام‌شده، بهترین بخش‌بندی در ده‌بار اجرا به‌عنوان بخش‌بندی انتخاب‌شده توسط روش پیشنهادی (GABP) در نظر گرفته شده و مقدار عددی مدل (زمان اجرا یا مصرف انرژی) توسط بخش‌بندی انتخاب‌شده، با میانگین مقدار عددی مدل (زمان اجرا یا مصرف انرژی) توسط بخش‌بندی‌های محاسبه‌شده در ده‌بار اجرا مقایسه شده است. در شرایط واقعی که بخش‌بندی در آغاز اجرای برنامه محاسبه می‌شود، می‌توان الگوریتم پیشنهادی (GABP) را به تعداد دفعات کمتری (مانند سه بار) اجرا و بهترین بخش‌بندی را انتخاب کرد.

(جدول-۴): WORGsهای تصادفی تولید شده

(Table-4): Generated Random WORGs

تعداد رأس	تعداد یال	تعداد رأس	تعداد یال
۲۰	۶۳	۵۰	۲۴۶
۳۰	۹۲	۶۰	۳۵۴
۴۰	۱۶۴	۷۰	۵۰۳

(جدول-۵): مقادیر پارامترهای مسأله

(Table-5): Problem Parameters

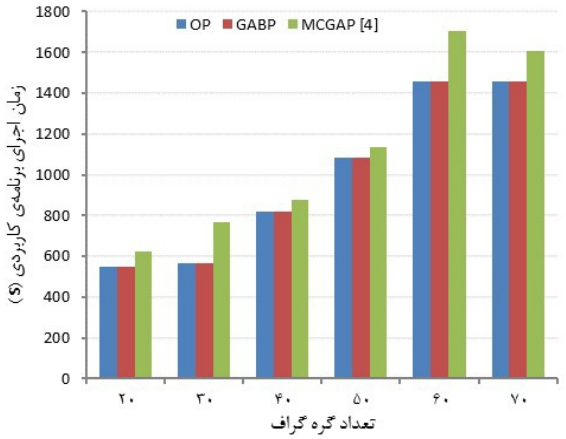
پارمتر	متغیر	مقدار
پهنای باند فعلی (لحظه‌ی اجرای برنامهٔ کاربردی)	b	100 (kB/sec)
پهنای باند کمینه	b_l	10 (kB/sec)
نسبت زمان اجرای یک نود روی موبایل به زمان اجرای آن روی سرور	k	5
توان cpu موبایل	P_{cpu}	0.9 (watt)
توان ارسال و دریافت واسط شبکه (wi-fi)	P_w	1.3 (watt)
ثابت تجربی	a	1.2

۵-۱- ارزیابی الگوریتم پیشنهادی در مقایسه با راه حل بهینه

در این بخش ابتدا عملکرد روش پیشنهادی از نقطه‌نظر زمان اجرا با برخی از روش‌های تطبیقی دیگر مقایسه می‌شود. یکی از نکات مهم در ارزیابی یک الگوریتم بخش‌بندی از جمله الگوریتم بخش‌بندی پیشنهادی بررسی کارایی آن از نقطه نظر زمان اجرا است؛ زیرا به‌دلیل فضای جست و جوی بزرگ این مسائل، برای پیدا کردن بخش‌بندی بهینه با درنظرگرفتن شرایطی مانند تطبیق با تغییرات پهنای‌باند و محدودیت تبادل داده، زمان بسیار زیادی مورد نیاز است. در جدول (۶) نتایج زمان اجرای الگوریتم پیشنهادی و الگوریتم‌های دیگر روی گراف‌های مختلف نشان داده شده است. همان‌طورکه در جدول مشاهده می‌شود هر دو الگوریتم تطبیقی GABP و MCGAP زمان اجرای بسیار کمتری نسبت به حالت بهینه دارند، البته زمان اجرای GABP اندکی بیشتر از MCGAP است که با این وجود مقدار زمان اجرا در حد معقول است. البته در قبال زمان اجرای بیشتر کارایی الگوریتم پیشنهادی که در نمودارهای بعدی این بخش دیده می‌شود، به‌مراتب از الگوریتم MCGAP بهتر است. البته با توجه به زمان اجرای

بسیار طولانی حالت بهینه در نتایج جدول (۶) مجبور به استفاده از گراف‌های با بیشینه اندازهٔ هفتاد گره بوده‌ایم.

در ادامهٔ این بخش میزان بهینگی الگوریتم GABP از نقطه‌نظر زمان اجرای برنامهٔ کاربردی مورد ارزیابی قرار می‌گیرد. به‌دلیل اینکه محاسبه بخش‌بندی تطبیقی بهینه برای گراف‌های بزرگ بسیار زمان‌بر است، بیشینه گراف‌هایی با اندازهٔ هفتاد گره ارزیابی شده است، که در این حالت بهترین جوابی که الگوریتم پیشنهادی GABP ارائه می‌کند با حالت بهینه برابر است که البته برای گراف‌های بزرگ‌تر طبیعتاً این امر نمی‌تواند صادق باشد. با این وجود همان‌طورکه در شکل (۳) مشاهده می‌شود، روش MCGAP که در مرجع [4] پیشنهاد شده در هیچ کدام از حالت‌ها بخش‌بندی تطبیقی بهینه را پیدا نکرده است. همچنین نزدیک‌بودن میانگین پاسخ‌های به‌دست‌آمده، توسط روش پیشنهادی در ده‌بار اجرا با حالت بهینه در جدول (۷)، نشان‌دهندهٔ دقت بالای الگوریتم GABP است.



(شکل-۳): زمان اجرای برنامهٔ کاربردی با اندازه گراف‌های مختلف در پهنای‌باند ۱۰۰ kB/s تحت بخش‌بندی‌های تطبیقی محاسبه شده

(Figure-3): Application Runtime for Various Graphs using Adaptive Partitioning

(جدول-۶): مقایسه‌ی زمان اجرای الگوریتم‌ها رویگراف‌های مختلف

(Table-6): Application Runtimes on Various Graphs

	۲۰	۳۰	۴۰	۵۰	۶۰	۷۰
OP	۰/۰۶۸۰	۳/۴۷۷۷	۲۰/۵۵۱۱	۹۷/۸۱۰۵	۳۳۹۴۳۰	۱۴۸۳۷۶
GABP	۰/۰۴۷۸	۰/۱۱۳۰	۰/۳۶۳۰	۰/۷۹۰۸	۱/۴۰۲۸	۲/۱۳۲۶
MCGAP	۰/۰۰۴۶	۰/۰۵۹۹	۰/۰۳۳۰	۰/۰۴۲۶	۰/۰۷۸۶	۰/۱۱۱۵

(جدول-۷): مقایسهٔ زمان اجرای برنامهٔ کاربردی توسط بخش‌بندی GABP در ده بار اجرا، با بخش‌بندی تطبیقی بهینه در پهنای‌باند ۱۰۰ kB/s روی گراف‌های مختلف

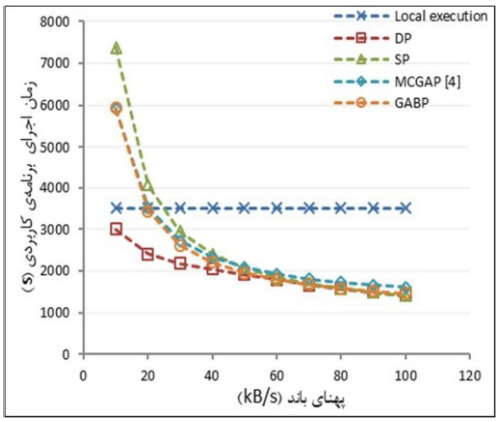
(Table-7): Comparison of Application Runtimes using GABP Partitioning for BW=100 kBps

	۲۰	۳۰	۴۰	۵۰	۶۰	۷۰
متوسط زمان اجرای برنامه کابردی توسط بخش‌بندی‌های GABP	۵۴۹/۷۹	۷۲۴/۸۹۶۹	۸۲۱/۲۹۸۲	۱۰۸۵/۸۹	۱۴۶۲/۴۲۷	۱۴۵۸/۸۵۸۹
زمان اجرای برنامه‌ی کابردی توسط بخش‌بندی تطبیقی بهینه	۵۴۹/۷۹	۷۲۴/۲۲	۸۲۱/۱۸۰۲	۱۰۸۶/۵۹۳	۱۴۵۵/۶۶	۱۴۵۷/۵۹

۲-۵- ارزیابی کارایی روش‌های مختلف از نقطه‌نظر زمان اجرا و مصرف انرژی برنامه کاربردی

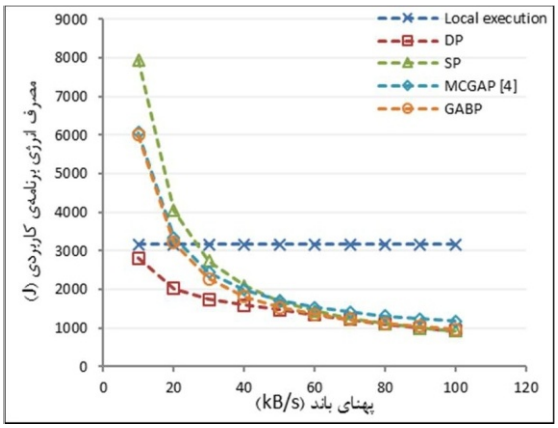
در این بخش کارایی روش‌های مختلف از نقطه‌نظر زمان اجرا و مصرف انرژی برنامهٔ کاربردی تحت بخش‌بندهای منتج، ارزیابی شده است. در شکل‌های (۴) و (۵) مشخص است که روش SP در پهنای‌باندهای پایین عملکرد خوبی ندارد و روش پیشنهادی (GABP) و روش MCGAP تطبیق بهتری با کاهش پهنای‌باند دارند و در این میان عملکرد روش پیشنهادی در پهنای باند، بالاتر از ۶۰ kB/s به عملکرد روش DP بسیار نزدیک‌تر است؛ روش DP اگرچه بهترین عملکرد را دارد؛ امّا محاسبهٔ بخش‌بندی در حین اجرا می‌تواند هزینه‌بر باشد و تغییر بخش‌بندی در حین اجرا ممکن است در بعضی از برنامه‌های کاربردی امکان‌پذیر نباشد. در بخش‌بندی تطبیقی، تعیین مناسب حد بالا موجب می‌شود از برون‌سپاری نودهایی که در پهنای‌باند به‌نسبه بالا موجب بهبود عملکرد می‌شوند، صرفه‌نظر شود؛ بدین ترتیب با وجود کاهش اندک عملکرد در پهنای‌باند بالا هزینهٔ ارتباطی نیز کاهش پیدا می‌کند و این کاهش هزینهٔ ارتباطی، تأثیر محسوسی در بهبود

عملکرد در پهنای‌باندهای پایین دارد. با این وجود جدول‌های (۴) و (۵) نشان می‌دهند هزینهٔ ارتباطی بالا موجب شده است که عملکرد Local execution (اجرای محلی برنامهٔ کاربردی روی تلفن همراه بدون بخش‌بندی) بهتر از عملکرد بخش‌بندی‌های ایجادشده توسط روش پیشنهادی (GABP) و الگوریتم MCGAP در پهنای‌باند ۱۰ kB/s باشد. بنابراین لازم است با درنظرگرفتن محدودیت تبادل داده در بخش‌بندی، هزینهٔ ارتباطی را بهتر کنترل کرد.



(شکل-۴): زمان اجرای برنامهٔ کاربردی در پهنای‌باندهای مختلف توسط بخش‌بندی‌های محاسبه‌شده

(Figure4): Application Runtime using Partitioning



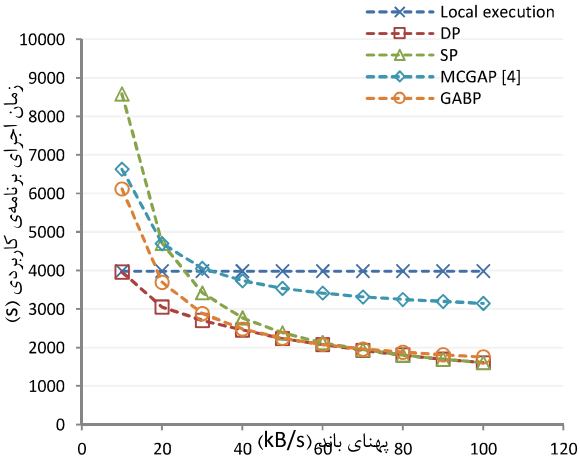
(شکل-۵): مصرف انرژی برنامهٔ کاربردی در پهنای‌باندهای مختلف توسط بخش‌بندی‌های محاسبه‌شده

(Figure-5): Energy Consmuption of Application using Partitioning

WORGs تصادفی با ۷۰ رأس و ۵۰۳ یال

Random WORGs using 70 Nodes and 503 Links

در ادامه ارزیابی دقیق‌تری از عملکرد روش پیشنهادی (GABP) و روش MCGAP [4] انجام گرفته است. در شکل (۶)، روی گراف تصادفی تولیدشده با هشتاد رأس و هفتصد یال، تحت بخش‌بندی محاسبه‌شده توسط روش پیشنهادی (GABP) نسبت به بخش‌بندی محاسبه‌شده توسط الگوریتم MCGAP زمان اجرای برنامهٔ کاربردی بسیار کمتر بوده است و اختلاف عملکرد دو الگوریتم نسبت به گراف‌های قبلی بسیار محسوس است. این مشاهده را می‌توان به این شکل توجیه کرد که همچنان‌که در جدول (۸) دیده می‌شود، کاهش تعداد گره‌های S_{b_1} و C_b در عملکرد الگوریتم MCGAP بسیار تأثیرگذار بوده و موجب افت کارایی در این الگوریتم شده است. در الگوریتم MCGAP که الگوریتمی حریصانه است، تنها تعداد $m = n - n_{C_b} - n_{s_{b1}}$ بخش‌بندی تولید می‌شود (درواقع m تعداد گره‌هایی هستند که باید بخش‌بندی شوند) که مبنای این بخش‌بندی نیز گره‌های S_{b_1} و C_b هستند.



(شکل-۶): زمان اجرای برنامهٔ کاربردی در پهنای‌باندهای مختلف توسط بخش‌بندی‌های محاسبه شده

(Figure-6): Application Runtime using Partitioning

(جدول-۸): تعداد گره‌های S_{b1} و C_b با توجه به مدل زمان اجرا و بر اساس گراف‌های تصادفی تولیدشده

(Table-8): Number of S_{b1} and C_b based on Random Graph Runtimes

نوع گراف	تعداد S_{b_1}	تعداد C_b	مجموع
گراف با ۳۰ گره	۱۴	۲	۱۶
گراف با ۴۰ گره	۲۴	۱	۲۵
گراف با ۵۰ گره	۳۱	۲	۳۳
گراف با ۶۰ گره	۳۰	۱	۳۱
گراف با ۷۰ گره	۴۳	۱	۴۴
گراف با ۸۰ گره	۱	۱	۲

در جدول (۹) دیده می‌شود که با وجود تبادل دادهٔ کمتر در بخش‌بندی ایجادشده توسط MCGAP، بخش‌بندی

محاسبه‌شده توسط روش پیشنهادی (GABP) در پهنای‌باند کمینه عملکرد بهتری داشته است؛ زیرا اختلاف در تعداد نودهای برون‌سپاری‌شده و میزان تبادل داده به‌گونه‌ای است که تأثیر کاهش زمان اجرا (ناشی از برون‌سپاری نودهای بیش‌تر)، بیش‌تر از تأثیر افزایش زمان تبادل داده بوده است. در جدول (۹) زمان اجرای برنامه کاربردی و متوسط زمان اجرای برنامهٔ کاربردی برای روش پیشنهادی (GABP)، به‌ترتیب براساس بهترین بخش‌بندی و میانگین زمان اجرا توسط کل بخش‌بندی‌های محاسبه‌شده در ده‌بار اجرا است. از آنجایی که الگوریتم MCGAP در هر اجرا یک بخش‌بندی را محاسبه می‌کند، زمان اجرای برنامهٔ کاربردی و متوسط زمان اجرای برنامهٔ کاربردی در جدول (۹) برای الگوریتم MCGAP برابر است.

(جدول-۹): زمان اجرای برنامهٔ کاربردی در پهنای باند 100 kB/s توسط بخش‌بندی‌های تطبیقی شکل (۴)

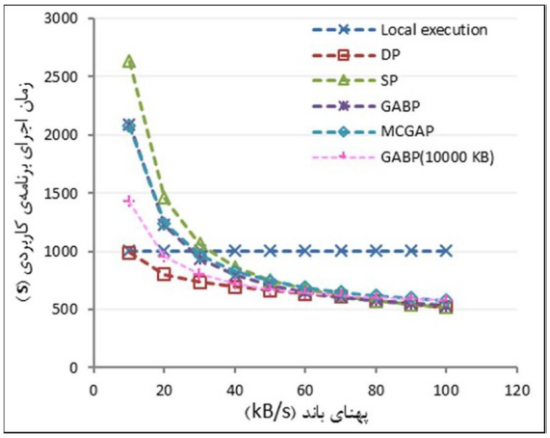
(Table-9): Application Runtime for BW=100 kBps using Adaptive Partitioning in Figure (4)

الگوریتم	زمان اجرای برنامه‌ی کاربردی (s)	متوسط زمان اجرای برنامه‌ی کاربردی (s)	متوسط زمان اجرای الگوریتم (s)	دادهٔ مبادله شده (KB)
GABP	۱۷۵۷/۶۹	۱۷۸۵/۲۵۹	۳/۸۱۰۵۱۷۹۳	۴۸۴۸۹
MCGAP	۳۱۵۳/۵۷	۳۱۵۳/۵۷	۰/۴۰۵۰۲۳۳	۳۸۶۳۷

همچنین جدول (۱۰) نشان می‌دهد که در بخش‌بندی GABP چهل گره برون‌سپاری شده‌اند که در بخش‌بندی MCGAP در وضعیت اجرای محلی قرار دارند. با میانگین زمان اجرای پنجاه ثانیه، برون‌سپاری چهل نود موجب کاهش زمان اجرا به میزان ۱۶۰۰ ثانیه می‌شود ((۵ ÷ ۵۰) – ۵۰) × ۴۰)، به این ترتیب با وجود اختلاف kB ۹۸۵۲ در میزان تبادل داده و بیشتربودن زمان تبادل داده به میزان ۹۸۵/۲ ثانیه در پهنای باند کمینه (۱۰ kB/s)، بخش‌بندی محاسبه‌شده توسط روش پیشنهادی (GABP) در مقایسه با بخش‌بندی MCGAP در پهنای‌باند کمینه نیز عملکرد بهتری داشته است.

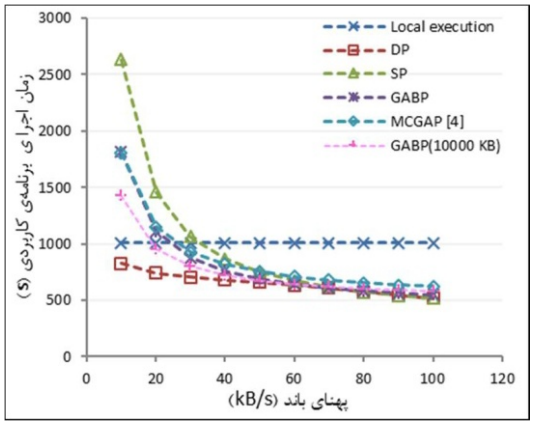
۳-۵- بررسی تأثیر خدمات وب و محدودیت تبادل داده

در این بخش تأثیر خدمات وب و محدودیت تبادل داده بر بخش‌بندی تطبیقی در روش پیشنهادی GABP و تأثیر خدمات وب بر بخش‌بندی انتخابی در سایر روش‌ها بررسی شده است. در شکل (۷) و شکل (۸) در نمودار GABP(10000 KB) بخش‌بندی تطبیقی به روش پیشنهادی با وجود محدودیت تبادل داده ۱۰۰۰۰ KB محاسبه شده است. در نمودارهای شکل (۸) و شکل (۱۰) گرهٔ هشت به‌عنوان خدمات وب در نظر گرفته شده است. مقایسهٔ نمودار DP در شکل‌های (۷) تا (۱۰) نشان می‌دهد که خدمات وب قراردادن گرهٔ هشت موجب افزایش زمان اجرای بهینه و انرژی مصرفی بهینه در پهنای‌باندهای پایین شده و در پهنای‌باندهای بالا تغییری ایجاد نکرده است.



(شکل-۷): زمان اجرای برنامهٔ کاربردی در پهنای‌باندهای مختلف توسط بخش‌بندی‌های محاسبه‌شده در عدم وجود خدمات وب

(Figure-7): Application Runtime Based on Partitioning for No-Web Service Case

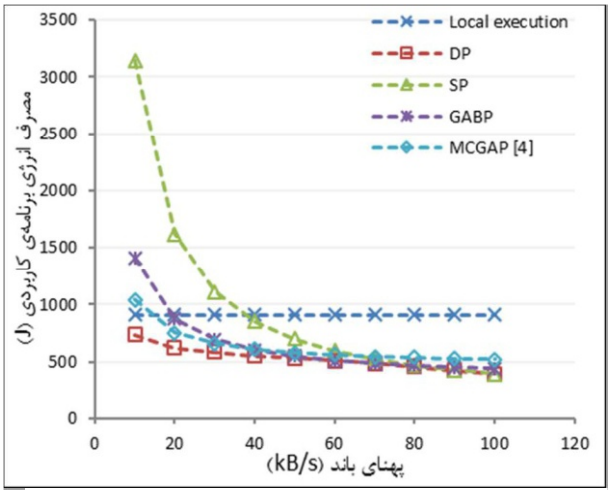


(شکل-۸): زمان اجرای برنامهٔ کاربردی در پهنای‌باندهای مختلف توسط بخش‌بندی‌های محاسبه‌شده با وجود خدمات وب

(Figure-8): Application Runtime Based on Partitioning for Web Service Case

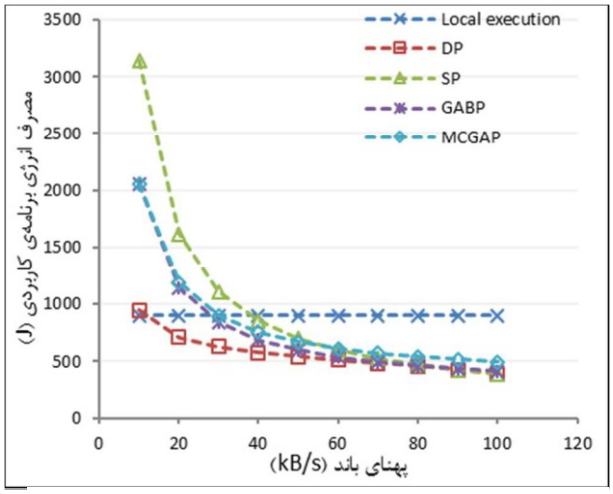
WORGs تصادفی با ۲۰ رأس و ۶۳ یال

Random WORGSs using 20 Nodes and 63 Links



(شکل-۹): مصرف انرژی برنامه کاربردی در پهنای‌باندهای مختلف توسط بخش‌بندی‌های محاسبه شده در عدم وجود خدمات وب

(Figure-9): Energy Consumption of Application Based on Partitioning for No-Web Service Case



(شکل-۱۰): مصرف انرژی برنامهٔ کاربردی در پهنای‌باندهای مختلف توسط بخش‌بندی‌های محاسبه شده با وجود خدمات وب

(Figure-10): Energy Consumption of Application Based on Partitioning for Web Service Case

WORGs تصادفی با ۲۰ رأس و ۶۳ یال

Random WORGs using 20 Nodes and 63 Links

(جدول-۱۰): بخش‌بندی‌های تطبیقی محاسبه‌شده در شکل (۴)

(Table-10): Adaptive Partitioning for Figure (4)

الگوریتم	بخش‌بندی
GABP	۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۰۱۰۱۰۰۰۰۰۱۰۰۰۰۰۰۰۱۰۰۰۰۰۰۰۰۰۰۰۰۰۱۱ ۰۰۰۰۰۰۰۰۰۰۱۱۰۰۱۱۰۱۰۱۰۰
MCGAP	۱۰۱۱۱۱۱۱۰۱۱۱۱۱۱۱۱۱۱۱۰۱۰۱۱۰۰۰۰۱۰۰۱۱۱۱۱۰۰۰۱۰۰۱۰۰۰۱۱۰۰۰۰۱۱۱۱۱۱ ۱۱۰۰۱۰۰۱۱۰۱۰۱۰۱۱۱۱۱۱۰۱

(جدول-۱۱): میزان دادهٔ مبادله‌شده توسط بخش‌بندی‌های محاسبه‌شده در شکل (۷) و شکل (۸)

(Table-11): Transferred Data using Claculated Partitioning Based on Figs 7 and 8

الگوریتم	عدم وجود خدمات وب	وجود خدمات وب
GABP	۱۴۰۳۹	۱۷۲۹۶
SP	۲۳۵۲۸	۲۳۵۲۸
MCGAP	۱۳۱۹۸	۱۶۶۱۵
GABP(10000 KB)	۹۴۵۸	۹۴۱۳

(جدول-۱۲): بخش‌بندی‌های بهنیه در پهنای‌بند ۴۰ kB/s و ۵۰ kB/s در شکل‌های (۷)، (۸)، (۹) و (۱۰)

(Table-12): Optimized Partitioning for Different Bandwidths Based on Figs 7, 8, 9, and 10

مدل	پهنای‌باند	وجود یا عدم وجود خدمات وب	بخش‌بندی	مقدار عددی مدل
زمان اجرا	۴۰	عدم وجود وب‌سرویس	۰۰۱۰۰۰۱۱۰۰۱۰۱۱۰۰۱۱۱۱	۶۸۱/۸۷۵
		وجود وب‌سرویس	۰۰۱۰۰۰۱۱۰۰۱۰۰۱۰۰۰۱۱۱	۶۹۵/۴۴۹۸
مصرف انرژی	۵۰	عدم وجود وب‌سرویس	۰۰۱۰۰۰۱۱۰۰۱۰۱۱۰۰۱۱۱۱	۵۳۳/۵۴۱۹
		وجود وب‌سرویس	۰۰۱۰۰۰۱۱۰۰۱۰۰۱۰۰۰۱۱۱	۵۴۱/۰۶۴

با افزایش زمان اجرا و انرژی مصرفی بهینه مقدار حد بالا نیز بیشتر می‌شود، جدول (۱۱) نشان می‌دهد این افزایش حد بالا باعث شده است که الگوریتم‌های تطبیقی در بخش‌بندی‌های محاسبه‌شده گره‌های بیشتری را برون‌سپاری کنند و درنتیجه دادهٔ بیشتری را نیز مبادله کنند، ازاین‌رو زمان اجرا و انرژی مصرفی تحت بخش‌بندی‌های محاسبه‌شده توسط MCGA و روش پیشنهادی (GABP) در شکل (۸) و شکل (۱۰) در مقایسه با شکل (۷) و شکل (۹) در پهنای‌باندهای بالا کاهش و در پهنای‌باندهای پایین به‌ویژه در پهنای‌باندهای ۱۰ kB/s، ۲۰ kB/s و ۳۰ kB/s به‌طور محسوسی افزایش یافته است. با توجه به اینکه زمان تبادل داده در مدل مصرف انرژی نسب به مدل زمان اجرا در pw (۱/۳ وات) ضرب می‌شود، تغییر در انرژی مصرفی بهینه در مقایسه با زمان اجرای بهینه بیشتر بوده است.

افزایش حد بالا در عملکرد روش پیشنهادی GABP با محدودیت تبادل داده ۱۰۰۰۰ KB (نمودار GABP(10000 (KB) تأثیری نداشته است، زیرا شرط محدود کننده همچنان محدودیت تبادل دادهٔ ۱۰۰۰۰ kB است. همچنین می‌توان مشاهده کرد در روش پیشنهادی با تعیین محدودیت تبادل داده و در نتیجه کنترل مستقیم هزینهٔ ارتباطی، بخش‌بندی تطبیقی با عملکرد بهتری ایجاد شده است.

جدول (۱۲) نشان می‌دهد که خدمات وب قراردادن گره ۸ در بخش‌بندی‌های بهینه تحت پهنای‌باند ۴۰ kB/s و ۵۰kB/s به‌ترتیب در مدل‌های زمان اجرا و مصرف انرژی موجب تغییر وضعیت گرهٔ ۴ (از اجرای محلی به برون‌سپاری) شده است.

۵-۴- بخش‌بندی و برون‌سپاری یک برنامهٔ کاربردی اندروید

در این بخش نحوهٔ بخش‌بندی و برون‌سپاری الگوریتم Stoer_Wagner را در قالب یک برنامهٔ کاربردی اندروید و با استفاده از ASP.NET Web API [17] تشریح شده است.

۵-۵- ساختن گراف

شکل (۱۱) گراف برنامهٔ کاربردی الگوریتم Stoer_Wagner را نشان می‌دهد. وزن رأس‌ها نشان‌دهندهٔ مرتبهٔ اجرایی و وزن یال‌ها نشان‌دهندهٔ میزان دادهٔ مبادله‌شده بین آن‌ها را بر اساس n و بر حسب بایت است، که n تعداد گره‌های گرافی است که الگوریتم روی آن اجرا می‌شود.

۵-۶- برون‌سپاری توسط Web API

برای انجام آزمون تجربی، الگوریتم Stoer_Wagner در قالب برنامهٔ کاربردی اندروید و ASP.NETWebAPI پیاده‌سازی شده و بخش‌بندی در سطح تابع انجام گرفته است، توابع برون‌سپاری‌شده با زبان C# بازنویسی شده و در قالب Web API پیاده‌سازی شده‌اند. برای فراخوانی یک تابع برون‌سپاری‌شده (Web API) یک درخواست POST HTTP به سرور ارسال می‌شود و نتایج نیز در قالب یک پاسخ HTTP به کلاینت (برنامهٔ کاربردی اندروید) باز می‌گردد.

۵-۷- نتایج آزمون تجربی بخش‌بندی و برون‌سپاری یک برنامهٔ کاربردی اندروید

در آزمایش انجام‌شده برنامهٔ کاربردی اندروید روی تلفن هوشمند ۱۶uawei honor 3c lite با مشخصات زیر اجرا شده است.Quad-Core Cortex-A7 CPU از نوع ۳۲ بیتی با فرکانس ۱/۳ GHs، حافظهٔ ۲ GB و سامانه‌عامل اندروید 4.4 KitKat. بخش‌های برون‌سپاری‌شده (Web APIها) نیز روی یک میزبان بارگذاری شده‌اند. برقرای ارتباط با سرور نیز از طریق Wi-Fi با نرخ انتقال دادهٔ ۵۱۲ kb/s (با نرخ واقعی حدود ۷۵ kB/s) انجام گرفته است. در آزمایش انجام‌شده حالت‌های زیر برای اجرای برنامهٔ کاربردی در نظر گرفته شده‌اند:

- در آزمایش انجام‌شده حالت‌های زیر برای اجرای برنامهٔ کاربردی در نظر گرفته شده‌اند:
- اجرای محلی: کل برنامهٔ کاربردی روی تلفن هوشمند اجرا شده است.
- بخش‌بندی ۱: گره‌های ۱و۲ (شکل (۱۱)) به‌صورت محلی اجرا شده‌اند و گره‌های دیگر نیز برون‌سپاری شده‌اند.
- بخش‌بندی ۲: گره‌های ۱، ۲، ۴ و ۶ (شکل (۱۱)) به‌صورت محلی اجرا شده‌اند و گرهٔ پنج برون‌سپاری شده‌است. با توجه با این که گرهٔ شش مرتبهٔ اجرایی پایینی دارد و وزن یال‌های متصل به این رأس زیاد است، این گره هم روی سرور و هم روی موبایل اجرا شده است تا موجب کاهش عملکرد نشود، این کار نیز می‌تواند ایدهٔ مناسبی برای پژوهش‌های آینده باشد (برخی از گره‌ها هم روی سرور و هم روی موبایل قابل اجرا باشند).

جدول (۱۳) زمان اجرای برنامهٔ کاربردی را در سه حالت با توجه به تعداد گره‌های گراف‌های تصادفی تولیدشده نشان می‌دهد. چهار گراف تصادفی با تعداد گره‌های ۵۰، ۱۰۰،

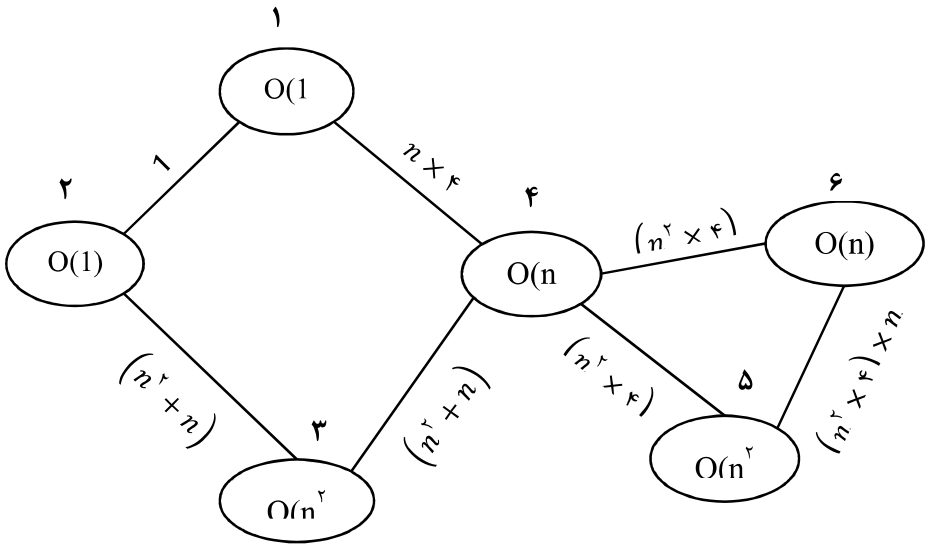
۱۸۰ و ۳۰۰ تولید شده و الگوریتم Stoer_Wagner روی هر گراف و برای هر یک از سه حالت در نظرگرفته شده، ده بار اجرا شده است. متوسط زمان تبادل داده شامل زمان برقراری ارتباط نیز می‌شود. نتایج نشان می‌دهد که اعمال بخش‌بندی ۲ کاهش عملکرد چشم‌گیری را در پی داشته است؛ البته این نتیجه قابل انتظار بود؛ زیرا این بخش‌بندی در مجموع موجب انتقال $50 \times (50^2 \times 4)$ بایت داده از موبایل به سرور شده است، به‌عبارت دیگر پنجاه درخواست HTTP و در هر درخواست $(50^2 \times 4)$ بایت داده به سرور ارسال شده است. به همین علت زمان تبادل داده و برقرای ارتباط بین موبایل و سرور به‌طور چشم‌گیری بالا بوده است. این در حالی است که با اجرای گرهٔ شش روی موبایل و سرور هزینهٔ انتقال، کاهش داده شده است.

نتایج نشان می‌دهد که برای تعداد گره‌های پنجاه و صد با اعمال کردن بخش‌بندی ۲ با وجود کاهش‌یافتن زمان اجرا روی تلفن همراه، تأخیر ایجادشده برای انتقادل داده و برقرای ارتباط با سرور قابل جبران نبوده، به همین علت اجرای برنامهٔ کاربردی به‌صورت محلی عملکرد بهتری داشته است. در شرایطی که زمان تبادل داده بیشتر از زمان اجرای محلی روی تلفن همراه است، برون‌سپاری اجرا، نمی‌تواند موجب بهبود عملکرد شود حتی اگر سرعت سرور بی‌نهایت باشد.

(جدول-۱۳): زمان اجرای برنامهٔ کاربردی در سه حالت مختلف اجرا با توجه به تعداد گره‌های گراف تصادفی تولیدشده

(Table-13): Application Runtime Based on Random Nodes Creation

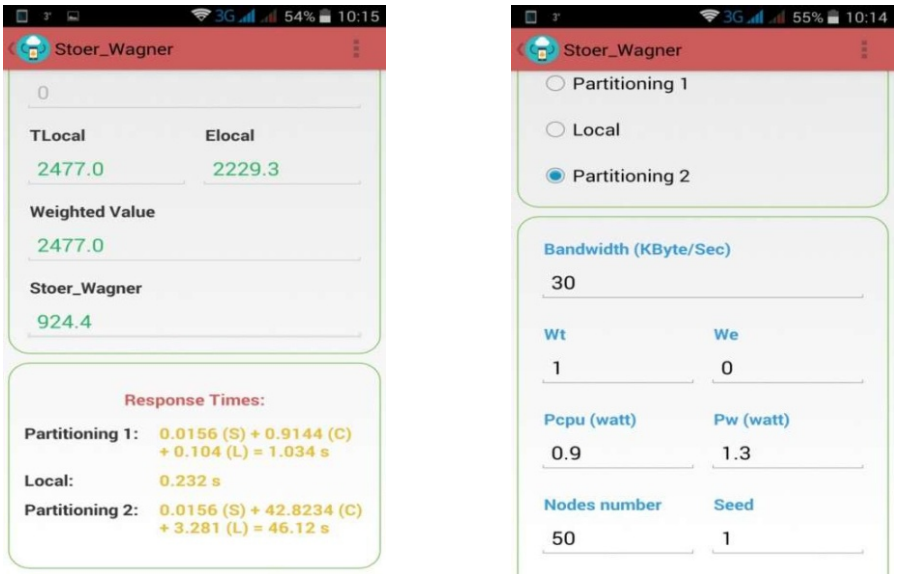
تعداد گره	حالت اجرا	متوسط زمان اجرا روی سرور	متوسط زمان تبادل داده	متوسط زمان اجرا روی موبایل	متوسط زمان اجرای برنامهٔ کاربردی
۵۰	بخش‌بندی ۱	۰/۰۰۷۸	۰/۹۱۳	۰/۱۱۰۵	۱/۰۳۱۳
	محلی	-	-	۰/۲۲۴۴	۰/۲۲۴۴
	بخش‌بندی ۲	۰/۰۳۱۲۱	۴۳/۵۹۲۸۹	۳/۳۹۰۸	۴۷/۰۱۴۹
۱۰۰	بخش‌بندی ۱	۰/۰۸۱۲۵	۱/۳۶۸۳۵	۰/۲۵۶۸	۱/۷۰۶۴
	محلی	-	-	۱/۰۴۹۸	۱/۰۴۹۸
۱۵۰	بخش‌بندی ۱	۰/۲۶۰۹۴	۲/۳۶۰۳۱	۰/۴۷۸۶	۳/۰۹۹۸۵
	محلی	-	-	۳/۰۸۹۶	۳/۰۸۹۶
۳۰۰	بخش‌بندی ۱	۲/۲۵۷۸۲	۵/۸۹۴۴۸	۱/۵۲۱۲	۹/۶۷۳۵
	محلی	-	-	۲۱/۵۵۹۴	۲۱/۵۵۹۴



(شکل-۱۱): گراف برنامهٔ الگوریتم Stoer_Wagner

(Figure-11): Stoer_Wagner Algorithm Graph

نتایج نشان می‌دهد که با بزرگ‌ترشدن گراف و افزایش تعداد گره‌ها، زمان اجرا روی تلفن همراه نیز بیشتر شده و عملکرد اجرای محلی و بخش‌بندی ۱ نیز به‌تدریج به یکدیگر نزدیک‌تر شده و در گراف با تعداد ۱۵۰ گره عملکرد آنها به‌طورتقریبی یکسان بوده است. با افزایش تعداد گره‌های گراف، میزان دادهٔ مبادله‌شده نیز بیشتر می‌شد؛ امّا نتایج نشان می‌دهد که بزرگ‌ترشدن گراف روی افزایش زمان اجرا روی تلفن همراه تأثیر بسیار بیشتری داشته، به‌طوری که با اعمال بخش‌بندی ۱ زمان اجرای الگوریتم Stoer_Wagner در گراف با سیصد گره نسبت به حالت اجرای محلی، حدود دوازده ثانیه کاهش یافته است. در شکل (۱۲) واسط کاربری (UI) برنامهٔ کاربردی پیاده‌سازی‌شده برای به‌دست‌آوردن بخش‌بندی بهینه با استفاده از الگوریتم Stoer_Wagner نشان داده شده است. برنامه کاربردی پیاده‌سازی‌شده در نشانی *http://offloading.somee.com* در قالب یک فایل فشرده قابل بارگیری است (کلمه عبور فایل فشرده: ۳۶).



(شکل-۱۲): پیاده‌سازی الگوریتم Stoer_Wagner در اندروید
(Figure-12): Android Implementation of Stoer_Wagner

۶- نتیجه‌گیری

زمان اجرا و انرژی مصرفی برنامه‌های کاربردی و در قلب آن‌ها، الگوریتم‌های مورد استفاده، در زمره چالش‌های قابل مطالعه در حوزهٔ محاسبات ابری قرار دارند. به‌سبب منابع سخت‌افزاری و نرم‌افزاری محدود، این مورد به‌صورت ویژه در تلفن‌های هوشمند و متعاقب آن محاسبات ابری سیار نمود بیشتری پیدا کرده است. به‌عنوان نمونه‌ای از الگوریتم‌های اشاره‌شده، به الگوریتم فیشر می‌توان اشاره کرد که برخلاف کاربرد بسیار در آشکارسازی سیگنال‌های صوتی و تصویری زمان اجرای بسیار بالایی را تحمیل برنامه می‌کند و راه‌حل‌های متعددی از جمله استفاده از فضاس پردازنده‌های گرافیکی برای رفع آن مورد مطالعه بوده است [18]. در این مقاله سه مدل بخش‌بندی، مدل بهینه‌سازی زمان اجرا، مدل بهینه‌سازی مصرف انرژی و مدل وزن‌دار بهینه‌سازی زمان اجرا و مصرف انرژی، با درنظرگرفتن تغییرات پهنای‌باند در تلفن هوشمند، استفاده از خدمات وب در برنامه‌های کاربردی و محدودیت تبادل داده ارائه شد و روشی مکاشفه‌ای مبتنی بر الگوریتم ژنتیک نیز برای برای محاسبهٔ بخش‌بندی با توجه به مدل‌های پیشنهای در زمان معقول ارائه شد. همچنین در یک آزمون تجربی نحوهٔ بخش‌بندی و برون‌سپاری الگوریتم بخش‌بندی Stoer_Wagner در قالب یک برنامهٔ کاربردی اندروید و ASP.NET Web API بیان شد.

نتایج شبیه‌سازی روی گراف‌های مختلف نشان داد که الگوریتم پیشنهادی با توجه به زمان اجرای بسیار پایین‌تر نسبت به روش Optimal-adaptive در گراف‌های کوچک قادر به پیدا کردن بخش‌بندی بهینه است و در گراف‌های بزرگ نیز عملکرد آن بهتر از الگوریتم حریصانهٔ (MCGAP) است و روش پیشنهادی تطبیق بهتری با تغییرات پهنای‌باند دارد (در بعضی از گراف‌ها تفاوت عملکرد آنها محسوس است)؛ همچنین نتایج نشان داد که استفاده از خدمات وب در برنامهٔ کاربردی با

افزایش میزان تبادل داده، می‌تواند موجب تغییر در بخش‌بندی بهینه در یک پهنای‌باند مشخص و تغییر بخش‌بندی تطبیقی با پهنای‌باند شود که با تعیین محدودیت تبادل داده و در نتیجه کنترل مستقیم هزینهٔ ارتباطی، می‌توان بخش‌بندی تطبیقی را با عملکردی بهتر ایجاد کرد.

نتایج آزمون تجربی نیز نشان داد اجزایی از برنامهٔ کاربردی که نیازمند محاسبات زیاد و تبادل دادهٔ کمی هستند، قابلیت بیشتری برای برون‌سپاری دارند. همچنین در آزمون تجربی انجام‌شده، مشاهده شد که با زیادشدن محاسبات در تلفن هوشمند، اختلاف زمان اجرای برنامهٔ کاربردی، در حالت اجرای محلی و اجرا روی سرور بیشتر می‌شود.

برای کارهای آینده پیشنهادهای زیر را مدّ نظر قرار داده‌ایم. بخش‌بندی یک برنامهٔ کاربردی اندروید با استفاده از مدل‌ها و الگوریتم بخش‌بندی پیشنهادی، ارائهٔ مدل‌هایی با درنظرگرفتن اجرای موازی اجزای برنامهٔ کاربردی و درنظرگرفتن حالتی که پاسخ بخش برون‌سپاری‌شده به تلفن هوشمند باز نگردد. قابل اجرابودن برخی از گره‌ها در هر دو بخش تلفن همراه و سرور که در بخش ۷-۳- به آن اشاره شد.

۷- مراجع 7- References

[1] K.Kumar and et al., "A survey of computation offloading for mobile systems", M*obile Networks and Applications*, vol. 18(1), p p. 129-140, 2013.

[2] X.Ma and et al., "When mobile terminals meet the cloud: computation offloading as the bridge", *IEEE Network*, vol. 27(5), pp. 28-33, 2013.

[3] A.R.Khan and et al., "A survey of mobile cloud computing application models", *Communications Surveys & Tutorials, IEEE*, vol. 16(1), pp. 393-413, 2014.

[4] J.Niu, W. Song, and M. Atiquzzaman, "Bandwidth-adaptive partitioning for distributed execution optimization of mobile applications", *Journal of Network and Computer Applications*, vol.37, pp. 334-347. 2014.

[5] K.Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*,vol. 43(4), pp. 51-56. 2010.

[6] B.-G.Chun and et al, "Clonecloud: elastic execution between mobile device and cloud", in *Proceedings of the sixth conference on Computer systems*, ACM, 2011.

[7] P. Di Lorenzo, S. Barbarossa, and S. Sardellitti, "Joint Optimization of Radio Resources and Code Partitioning in Mobile Cloud Computing", arXiv preprint arXiv:1307, 2013, pp3835.

[8] F.Xia and et al, "Phone2Cloud: Exploiting computation offloading for energy saving on smartphones in mobile cloud computing", *Information Systems Frontiers*, vol. 16(1), pp. 95-111. 2014

[9] M.H. Tandel and V.S. Venkitachalam, "Cloud Computing in Smartphone: Is offloading a better-bet?", CS837-F12-MW-04A Wichita State University, 2013.

[10] T.Verbelen and et al, "Graph partitioning algorithms for optimizing software deployment in mobile cloud computing", *Future Generation Computer Systems*, vol. 29(2), pp. 451-459. 2013.

[11] E.Cuervo and et al , "MAUI: making smartphones last longer with code offload", in *Proceedings of the 8th international conference on Mobile systems, applications, and services.* ACM, 2010.

[12] Y.Zhang and et al, "Refactoring android java code for on-demand computation offloading", in *ACM SIGPLAN Notices* .ACM, 2012.

[13] R.Kemp and et al, "Cuckoo: a computation offloading framework for smartphones", in *Mobile Computing, Applications, and Services*, Springer, pp. 59-79, 2012.

[14] D.Kovachev, T. Yu, and R. Klamma, "Adaptive computation offloading from mobile devices into the cloud", in *Parallel and Distributed Processing with Applications (ISPA)*, *IEEE 10th International Symposium on*, 2012.

[15] X.Wei and et al, "MVR: An Architecture for Computation Offloading in Mobile Edge Computing", in *Edge Computing (EDGE)*, IEEE International Conference on. 2017.

[16]http://developer.android.com/guide/components/services.html (last accessed 12 Apr 2016).

[17]https://msdn.microsoft.com/enus/library/dd203052.aspx.

[18] h.sadeghi and A. Akhavan Bitaghsir, "Signal Detection Based on GPU-Assisted Parallel Processing for Infrastructure-based Acoustical Sensor Networks", *Signal and Data Processing*, vol.14(4), p p. 19-30. 2018.

سیاوش زاهدی فارغ‌التحصیل دوره کارشناسی ارشد رشته مهندسی فناوری اطلاعات از دانشگاه صنعتی ارومیه است. حوزه تخصصی وی بخش‌بندی پرداش و برون سپاری محاسبات در محیطهای ابری است. ایشان همچنین در حوزه توسعه برنامه‌های کاربردی مبتنی بر ویندوز، وب (asp.net mvc) و تلفن همراه (بر پایه اندروید) فعال بوده و برنامه‌های بسیاری تولید کرده است.

نشانی رایانامه ایشان عبارت است از:

syavash.36@gmail.com

صالح یوسفی دارای مدارک کارشناسی مهندسی کامپیوتر سخت‌افزار، کارشناسی ارشد مهندسی کامپیوتر معماری کامپیوتر و دکترای مهندسی کامپیوتر شبکه‌های کامپیوتری، هر سه از دانشگاه علم و صنعت ایران و در حال حاضر دانشیار گروه مهندسی کامپیوتر دانشگاه ارومیه است. حوزه‌های تخصصی ایشان شامل سامانه‌های مخابراتی هوشمند وسائل نقلیه، شبکه‌های تلفن همراه و بی‌سیم، محاسبات ابری، و مدل‌های اقتصادی در شبکه‌ها است.
نشانی رایانامه ایشان عبارت است از:
youssefi@cc.iut.ac.ir

وحید سلوک استادیار و مدیر گروه مهندسی فناوری اطلاعات و رئیس پردیس دانشگاهی دانشگاه صنعتی ارومیه و دارای مدرک دکترای مخابرات و شبکه از دانشگاه سلطنتی مالزی است. حوزه تخصصی ایشان شامل مدیریت حرکت، تخصیص منابع و کدگذاری شبکه‌های بیسیم و تلفن همراه و شبکه‌های تلفن همراه نامتنجانس، محاسبات ابری تلفن همراه، کیفیت سرویس و کیفیت تجربه در شبکه‌های سلولی، و بهینه‌سازی مصرف انرژی در شبکه‌های بی‌سیم است.
نشانی رایانامه ایشان عبارت است از:
v.solouk@it.uut.ac.ir