

## تخصیص وظایف چندهدفه در واگذاری به ابر سیار

سیمین قاسمی فلاورجانی<sup>۱</sup>، کارشناسی ارشد، محمدعلی نعمت‌بخش<sup>۲</sup>، دانشیار، بهروز شاهقلی قهفرخی<sup>۳</sup>، استادیار

۱- گروه مهندسی نرم‌افزار- دانشکده مهندسی کامپیوتر - دانشگاه اصفهان - اصفهان - ایران - simin.ghasemi@gmail.com

۲- گروه مهندسی نرم‌افزار - دانشکده مهندسی کامپیوتر - دانشگاه اصفهان - اصفهان - ایران - nematbakhsh@eng.ui.ac.ir

۳- گروه مهندسی فناوری اطلاعات - دانشکده مهندسی کامپیوتر - دانشگاه اصفهان - اصفهان - ایران - shahgholi@eng.ui.ac.ir

چکیده: در رایانش ابری سیار، قدرت محاسباتی دستگاه‌های سیار با واگذاری بخش‌هایی از نرم‌افزار به ابر، افزایش می‌یابد. تخصیص وظایف بین دستگاه‌های سیار مجاور که نقش تأمین‌کنندگان منابع را بر عهده دارند، یکی از مسائل مهم و چالش‌برانگیز در رایانش ابری سیار است. هدف از تخصیص وظایف در واگذاری یک نرم‌افزار، کمینه‌سازی دو معیار زمان کل اجرای برنامه و انرژی مصرفی دستگاه‌های سیار شرکت‌کننده در واگذاری و همچنین تأمین قیود کیفیت سرویس موردنظر برنامه کاربردی می‌باشد. در این مقاله، مسأله تخصیص وظایف در ابر سیار به صورت یک بهینه‌سازی چندهدفه قیددار مدل‌سازی و روشی دو مرحله‌ای برای حل آن پیشنهاد شده است. در مرحله اول، برای کسب مجموعه جواب‌های پارتو، یک الگوریتم مبتنی بر شاخه و حد چندهدفه (MRABB) طراحی شده است. در مرحله دوم، با استفاده از روش رتبه‌بندی ترجیحات از طریق شباهت به جواب ایده‌آل (TOPSIS) و گرفتن ترجیحات کاربر، بهترین جواب مصالحه‌گر تعیین می‌گردد. روش پیشنهادی روی بستر واگذاری به ابر سیار (به اختصار OMC)، توسعه داده شده که وظیفه جمع‌آوری اطلاعات زمینه‌ای مورد نیاز تصمیم‌گیرنده و مدیریت کل فرایند واگذاری را بر عهده دارد. نتایج آزمایش‌ها نشان می‌دهد که روش پیشنهادی در مقایسه با الگوریتم‌های تخصیص وظایف پیشین، به خوبی مصالحه میان زمان و انرژی را مدیریت می‌کند.

واژه‌های کلیدی: رایانش ابری سیار، واگذاری، تخصیص وظایف، بهینه‌سازی چندهدفه، آگاهی از زمینه.

## Multi-Objective Task Allocation in Offloading to Mobile Cloud

S. Ghasemi-Falavarjani, M.Sc.<sup>1</sup>, M.A. Nematbakhsh, Associate professor<sup>2</sup>, B. Shahgholi Ghahfarokhi, Assistant professor<sup>3</sup>

1- Department of Software Engineering, University of Isfahan, Isfahan, Iran, Email: simin.ghasemi@gmail.com

2- Department of Software Engineering, University of Isfahan, Isfahan, Iran, nematbakhsh@eng.ui.ac.ir

3- Department of Information Technology Engineering, University of Isfahan, Isfahan, Iran, Email: shahgholi@eng.ui.ac.ir

**Abstract:** Mobile Cloud Computing (MCC) augments capabilities of mobile devices by offloading applications to the cloud. Task allocation is one of the most challenging issues in MCC considering neighboring mobile devices as the service providers. Given an application offloading request, the objective of the task allocation is to select service providers minimizing the completion time of the application offloading as well as consumed energy of all participating mobile devices while satisfying some QoS constraints. This paper models the task allocation problem in MCC as a constrained multi-objective optimization and proposes a two stages approach to solve the problem. In the first stage, a Multi-objective Resource Allocation based on Branch and Bound algorithm (MRABB) is designed to obtain Pareto solution set. In the subsequent stage, using Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) method and given user's preferences, the best compromise solution is determined. Furthermore, a context-aware software for Offloading in Mobile Cloud (OMC) is designed and implemented to collect contextual used in task allocation, and manage the offloading process. The results show the ability of the proposed multi-objective task allocation method to manage the trade-off between time and energy comparing to traditional algorithms.

**Keywords:** Mobile cloud computing, offloading, resource allocation, multi-objective optimization, context-awareness.

تاریخ ارسال مقاله: ۹۴/۸/۱۶

تاریخ اصلاح مقاله: ۹۴/۱۲/۸

تاریخ پذیرش مقاله: ۹۵/۲/۱۱

نام نویسنده مسئول: محمدعلی نعمت‌بخش

نشانی نویسنده مسئول: ایران - اصفهان - دانشگاه اصفهان - دانشکده مهندسی کامپیوتر

## ۱- مقدمه

با وجود پیشرفت‌های قابل توجه در فناوری‌های دستگاه‌های سیار، هنوز دستگاه‌های سیار، قادر به اجرای نرم‌افزارهای با پردازش سنگین (با کیفیت سرویس قابل قبول کاربر) مانند پردازش صوت و تصویر، داده‌کاوی و شبکه‌های اجتماعی نیستند. برای حل مشکل محدودیت منابع دستگاه‌های سیار (انرژی محدود باتری، قدرت کم پردازنده)، واگذاری<sup>۱</sup> بخش‌هایی از پردازش نرم‌افزارها، از دستگاه سیار به ابر پیشنهاد شده است. به این ترتیب، رایانش ابری سیار توانایی دستگاه‌های سیار برای اجرای نرم‌افزارها را افزایش می‌دهد [۱].

در رایانش ابری سیار یک معماری سه‌لایه شامل سرورهای ابر راه دور در اینترنت، سرورهای محاسباتی محلی (cloudlet) و دستگاه‌های سیار مجاور تعریف شده است. اما منابع ابر راه دور همیشه در دسترس نبوده و هزینه استفاده از آن‌ها بالاست [۲] و از طرف دیگر، استفاده از cloudlet محلی، قابلیت جابجایی کاربران دستگاه‌های سیار را محدود کرده و به زیرساختار از قبل آماده شده نیاز دارد [۱، ۵-۳]. به همین دلیل، در این مقاله، لایه سوم رایانش ابری سیار شامل دستگاه‌های سیار در همسایگی کاربر (ابر سیار)، در نظر گرفته می‌شود. در ابر سیار، کاربر و تأمین کنندگان سرویس هر دو دستگاه سیار می‌باشند.

در رایانش ابری سیار، تخصیص وظایف بین دستگاه‌های سیار مجاور یکی از مسائل چالش برانگیز می‌باشد. در این مقاله، هدف از تخصیص وظایف در واگذاری یک نرم‌افزار در ابر سیار، کمینه‌سازی دو معیار زمان کل اجرا برنامه و انرژی مصرفی دستگاه‌های سیار شرکت کننده در واگذاری است به طوری که برخی قیود کیفیت سرویس مورد نظر برنامه کاربردی نیز تأمین گردد. علاوه بر این، برای تخصیص وظایف دقیق، پویا و سازگار، لازم است اطلاعات زمینه‌ای دستگاه‌های سیار مجاور (مانند قدرت پردازش و سطح انرژی)، برنامه کاربردی (مانند مدت زمان اجرا) و محیط (مانند پهنای باند شبکه) جمع‌آوری شوند. همچنین، به یک سیستم برای یکپارچه‌سازی و مدیریت مراحل فرایند واگذاری در ابر سیار نیاز است. از این رو، بستر واگذاری به ابر سیار<sup>۲</sup> (به اختصار OMC) برای انجام فرایند واگذاری توسعه داده شده است.

در سال‌های اخیر، چندین بررسی جامع در بحث واگذاری در رایانش ابری سیار ارائه شده است [۱]. از بین کارهای متعدد انجام شده در این زمینه، تعداد محدودی تحقیق، مسئله تخصیص وظایف در ابر سیار و تعیین سرویس دهنده مناسب از میان دستگاه‌های سیار مجاور را برای واگذاری هر یک از وظیفه‌ها مورد بررسی قرار داده‌اند [۴، ۹-۶]. برای مثال، اسکاوتنجر<sup>۳</sup>، سیستم تأمین کننده سایبری<sup>۴</sup> و یک زمان بند مبتنی بر پروفایل دوگانه (شامل دو پروفایل گره محور و وظیفه محور) برای تعیین سرویس دهنده مناسب ارائه کرده است [۶]. در زمان بند اسکاوتنجر، وظیفه‌ها بر اساس سرعت ماشین‌های مقصد به آن‌ها انتساب داده می‌شود تا زمان پاسخ کل کمینه شود. اما مقاله فوق تنها زمان اجرا را به عنوان معیار تخصیص وظایف بررسی کرده و انرژی مصرفی دستگاه‌های سیار را در نظر نگرفته است.

در مرجع [۴]، یک طرح واگذاری آگاه به زمینه برای محیط‌های نظیر به نظیر سیار و به اشتراک گذاری وظیفه‌ها بین دستگاه‌های مجاور از طریق واگذاری ارائه شده است. در این طرح، اطلاعات دستگاه‌های سیار مجاور و محیط، جمع‌آوری شده و با توجه به سیاست‌هایی تعیین شده برای واگذاری آگاه به زمینه، از اطلاعات زمینه‌ای برای استنتاج جنبه‌های مورد نیاز در فرایند واگذاری استفاده می‌شود. یکی از سه هدف زیر در تخصیص وظایف ارائه شده در این رساله برای انتخاب ماشین مقصد در نظر گرفته می‌شود:

- کمینه‌سازی انرژی مصرفی دستگاه سیار مبدأ
- کمینه‌سازی زمان اجرای برنامه کاربردی
- بیشینه‌سازی موفقیت‌آمیز بودن فرایند واگذاری با توجه به حرکت دستگاه‌های سیار و میزان اعتماد به آن‌ها

همچنین، برای تخصیص وظایف یک الگوریتم پویا و وفق‌پذیر مبتنی بر شاخه و حد ارائه شده است. اما در مسئله تخصیص وظایف در پژوهش فوق، تنها یکی از سه هدف زمان، انرژی مصرفی یا موفقیت‌آمیز بودن واگذاری در نظر گرفته شده و این سه معیار همزمان با هم بررسی نشده است. همچنین در کمینه‌سازی انرژی مصرفی، تنها انرژی مصرفی دستگاه مبدأ مدنظر قرار گرفته و منابع مصرفی سایر دستگاه‌های شرکت کننده در واگذاری، نادیده گرفته شده است. به علاوه، در کمینه‌سازی زمان اجرای برنامه کاربردی، هدف مقاله ذخیره‌سازی میزان استفاده از CPU در دستگاه مبدأ بر اثر واگذاری هر چه بیش تر بخش‌های برنامه بوده و زمان کل اجرا را به عنوان هدف بهینه‌سازی کمینه نکرده است.

شی و همکارانش، سیستم سرنديپیتی<sup>۵</sup> را طراحی و پیاده‌سازی کرده‌اند که دستگاه سیار مشتری را قادر می‌سازد که محاسبات خود را به دستگاه‌های سیار موجود در محیط اطرافش که به صورت متناوب با یکدیگر دیدار می‌کنند، واگذار کند [۸]. نویسندگان با فرض یکسان بودن وظایف، دو نسخه الگوریتم زمان بندی حریمانه بنام‌های سرنديپیتی بهینه‌ساز زمان و سرنديپیتی آگاه از انرژی را ارائه کرده‌اند. در الگوریتم بهینه‌ساز زمان، برای هر وظیفه، گره مقصد با توانایی اجرا در کمترین زمان انتخاب می‌شود. اما هدف الگوریتم آگاه از انرژی، افزایش طول عمر شبکه و تأمین شرط زمانی برای کل برنامه کاربردی می‌باشد. این الگوریتم، برای هر وظیفه، گره با کمترین مقدار انرژی (مطابق مدل انرژی [۱۰]) را انتخاب می‌کند، به طوری که شرط زمانی واگذاری نیز ارضا شود. همان طور که در مقاله [۸] ذکر شده است، این الگوریتم تخمین خوبی از بهینه‌آل فراهم می‌کند، ولی الزاماً میانگین انرژی مصرفی را کمینه نمی‌کند. همچنین اگرچه این مقاله تأثیر دو معیار زمان و انرژی در تخصیص وظایف در واگذاری به دستگاه‌های سیار مجاور را بررسی کرده است، ولی دو معیار زمان و انرژی را با هم در تصمیم‌گیری بهینه نمی‌کند.

مقاله [۹] یک محیط ابر متشکل از دستگاه‌های سیار (MDC) را در نظر گرفته که شامل دستگاه‌های سیاری است که با یکدیگر کارهای

یک روش دو مرحله‌ای برای حل آن پیشنهاد گردیده است. در مرحله اول، برای به دست آوردن مجموعه جواب‌های پارتو، با توسعه الگوریتم تحقیق قبلی، یک الگوریتم تخصیص منابع چندهدفه مبتنی بر شاخه و حد چندهدفه (MRABB)<sup>۹</sup> طراحی شده است. در مرحله دوم، برای تصمیم‌گیری چندمعیاره و انتخاب جواب نهایی مصالحه‌گر و همچنین رفع مشکل استفاده از SAW در مقاله قبلی، از روش رتبه‌بندی ترجیحات از طریق شباهت به جواب ایده‌آل (TOPSIS)<sup>۱۰</sup> استفاده شده است که با ویژگی‌های مسئله این مقاله، تطابق دارد و با گرفتن ترجیحات کاربر، بهترین جواب مصالحه‌گر تعیین می‌گردد. علاوه بر این، با توسعه OMC، یک نرم‌افزار واگذاری در ابر سیار، موسوم به OMC<sup>۱۱</sup>، برای جمع‌آوری اطلاعات زمینه‌ای موردنیاز در تصمیم‌گیری برای تخصیص وظایف و مدیریت کل فرایند واگذاری طراحی شده است. همچنین با توسعه آزمایش‌ها، الگوریتم پیشنهادی در شرایط مختلف و گسترده مورد ارزیابی قرار داده شده و سربار ناشی از اجرای الگوریتم پیشنهادی در مقایسه با الگوریتم‌های قبلی نیز بررسی گردیده است.

ساختار مقاله در ادامه به این صورت تنظیم شده است: در بخش ۲ مسئله تخصیص وظایف در ابر سیار فرموله شده و یک روش پیشنهادی دو مرحله‌ای برای حل آن ارائه شده است. در بخش ۳، معماری نرم‌افزار OMC، فرایند واگذاری و تخمین پروفایل‌ها توضیح داده شده‌اند. بخش ۴، نتایج ارزیابی روش تخصیص وظایف پیشنهادی را تشریح کرده و در پایان نتیجه‌گیری و کارهای آینده ارائه شده است.

## ۲- تخصیص وظایف در ابر سیار

مسئله تحقیقاتی این مقاله، چگونگی تخصیص وظایف برای یک درخواست سرویس واگذاری در ابر سیار است به طوری که دو معیار زمان اتمام واگذاری و انرژی مصرفی دستگاه‌های سیار، هم‌زمان کمینه شده و برخی قیود کیفیت سرویس از جمله مهلت زمانی اجرا و حداقل انرژی باقی‌مانده دستگاه‌های سیار نیز تأمین گردد.

مفروضات در نظر گرفته‌شده در مدل پیشنهادی برای مسئله تخصیص وظایف عبارت‌اند از: (۱) در هر زمان، فقط یک واگذاری در حال اجرا است؛ (۲) هنگام واگذاری، دستگاه‌های سیار برنامه دیگری در حال اجرا ندارند؛ (۳) دستگاه‌های سیار تا پایان فرایند واگذاری، در محدوده ارتباطی یکدیگر باقی می‌مانند؛ (۴) ارتباط دستگاه‌های سیار به صورت مستقیم (یک گامی) و از طریق شبکه Wi-Fi است؛ (۵) دستگاه‌های سیار همگی درستکار بوده و با یکدیگر همکاری می‌کنند؛ (۶) بخش‌بندی برنامه از قبل مشخص شده است و زیربرنامه‌های قابل واگذاری، مستقل بوده و قابلیت اجرا به صورت موازی را دارند؛ (۷) زیربرنامه‌های برنامه کاربردی یکسان و شبیه هم بوده و بر اساس آن، مسئله تخصیص وظایف مدل می‌شود.

### ۲-۱- مدل تخصیص وظایف در ابر سیار

فرض کنید برنامه‌ای که کاربر درخواست واگذاری آن را دارد شامل  $n$  عدد زیربرنامه مستقل، یکسان، و قابل اجرا به صورت موازی است.

مشترک و تعاملات زیادی دارند. سیاست تخصیص منبع در این مقاله، بیشینه کردن طول عمر گره‌ها در محیط MDC است. همچنین نویسندگان مقاله، یک سکوی مشابه با سرنویستی [۸] پیاده‌سازی کرده و از آن به عنوان یک بستر آزمایشی برای ارزیابی مدل‌های محاسباتی و پروفایل‌های مختلف و بررسی کارایی الگوریتم تخصیص منبع پیشنهادی خود در افزایش طول عمر MDC استفاده کرده‌اند. اما زمان اتمام وظیفه‌ها که یکی دیگر از معیارهای مهم کیفیت سرویس در واگذاری است، در این مقاله در نظر گرفته نشده است.

تحقیقات پیشین در تخصیص وظایف در ابر سیار، تنها یکی از دو معیار زمان یا انرژی مصرفی واگذاری را به عنوان معیار بهینه‌سازی بررسی کرده و هیچ کدام، دو معیار زمان و انرژی را به طور هم‌زمان در نظر نگرفته‌اند، در حالی که هر دو معیار در تعیین مقصد وظیفه‌ها تأثیرگذار هستند. زمان اتمام اجرا در بسیاری از برنامه‌ها اهمیت بالایی داشته و به علاوه از نظر کاربر، یکی از معیارهای مهم کیفیت سرویس می‌باشد. از طرف دیگر، با توجه به انرژی محدود باتری در دستگاه‌های سیار، انرژی مصرفی این وسایل حین اجرای برنامه‌ها نیز حائز اهمیت است. این در حالی است که دو معیار زمان و انرژی مصرفی معمولاً در تقابل با یکدیگر<sup>۶</sup> می‌باشند، به خصوص اگر دستگاه‌های سیار ناهمگن باشند، یعنی کاهش یکی باعث افزایش دیگری خواهد شد. مثلاً برای کاهش زمان اجرا لازم است وظیفه‌ها به دستگاه‌هایی که سریع هستند ولی مصرف انرژی بالایی دارند، واگذار شود که مصرف انرژی کل شبکه را افزایش می‌دهد. بنابراین کمینه کردن هر دو معیار زمان و انرژی، مسئله تخصیص وظایف را به یک بهینه‌سازی دو معیاره تبدیل می‌کند که جزء مسائل ان پی - سخت به‌شمار می‌رود.

در پژوهش پیشین نویسندگان [۱۱]، مسئله تخصیص وظایف در ابر سیار بررسی شده و دو معیار زمان و انرژی مصرفی هم‌زمان بهینه شده است. همچنین یک چارچوب واگذاری به ابر سیار (موسوم به OMC<sup>۷</sup>) برای مدیریت مراحل واگذاری ارائه شده است. در تحقیق مذکور، برای حل مسئله بهینه‌سازی چندهدفه تخصیص منابع، یک الگوریتم تخصیص منابع چندمعیاره بهینه و عادلانه مبتنی بر شاخه و حد طراحی گردیده است. این الگوریتم مجموعه جواب‌های پارتو را به دست می‌دهد و برای انتخاب جواب نهایی مصالحه‌گر<sup>۸</sup>، از روش جمع وزن دار ساده (SAW) استفاده می‌کند. SAW یکی از روش‌های دسته‌بندی امتیازدهی در تصمیم‌گیری چندمعیاره است که جمع وزن دار معیارها را به عنوان تابع امتیاز محاسبه می‌کند. در این روش باید معیارها مستقل باشند یعنی افزایش یک معیار باعث کاهش معیار دیگر نشود [۱۲]. ولی از آنجایی که در مسئله تخصیص وظایف موردبررسی در این مقاله، دو معیار زمان و انرژی وابسته به یکدیگر هستند و ممکن است کاهش زمان اجرا باعث افزایش مصرف انرژی گردد (به خصوص در سیستم‌های ناهمگن)، بنابراین روش SAW مناسب این موضوع نمی‌باشد.

در این تحقیق که توسعه پژوهش [۱۱] است، مدل تخصیص وظایف در ابر سیار به صورت یک بهینه‌سازی چندهدفه قیددار، توسعه یافته و

$$tl_j = \begin{cases} b_j * t_j + \sum_{k=2}^{m+1} sendDelay_k, & j = 1 \\ b_j * \left[ t_j + \left( \frac{Vin + Vout}{BW_j} \right) \right] + Qdelay_j, & j > 1 \end{cases} \quad (1)$$

$$sendDelay_k = (n - b_0) * \left( \frac{Vin}{BW_k} \right)$$

$$Qdelay_j = \sum_{l=2}^{j-1} b_l * \left( \frac{Vin}{BW_l} \right)$$

از آنجایی که کانال ارتباطی و پهنای باند دستگاه سیار مبدأ با هر دستگاه سیار دیگر متفاوت است ( $BW_j$ ), زمان ارسال به هر گره سرویس دهنده متفاوت خواهد بود. این مدت زمان در رابطه (۱) با  $sendDelay_k$  نشان داده شده است. البته در اینجا، تعداد گره‌های سیار پیرامون اندک و تفاوت پهنای باند ناچیز است. بنابراین، با فرض اینکه در هر لحظه فقط گره مبدأ در حال ارسال داده است و بقیه گره‌ها انتقال داده‌ای ندارند، در آزمایش‌ها از متوسط پهنای باند استفاده می‌شود. شایان ذکر است که ارسال داده‌ها در دستگاه سیار مشتری، در اجرای وظیفه‌های محلی تأخیر ایجاد می‌کند.

هر گره سرویس دهنده ( $R_j$ ) ( $j > 1$ ) وظیفه‌های انتسابی خود را در زمان  $tl_j$  تکمیل می‌کند که متأثر از سه عامل است: (۱) مجموع زمان اجرای وظیفه‌ها روی آن گره ( $R_j$ )؛ (۲) زمان انتقال اطلاعات مرتبط با وظیفه‌ها؛ (۳) تأخیر صف ناشی از ارسال فایل‌ها در دستگاه سیار مشتری. مولفه زمان بند ارسال فایل‌ها در دستگاه مشتری، فایل وظیفه‌ها را به ترتیب قدرت پردازش گره‌های مقصدشان ارسال می‌کند. بنابراین مدت زمانی که فایل هر وظیفه در صف دستگاه مشتری منتظر می‌ماند تا به گره‌ی مقصدش ارسال شود، برابر با مجموع زمان انتقال فایل‌هایی است که باید قبل از آن به گره‌های دیگر ارسال شوند ( $Qdelay_j$  زمان انتظار سرویس دهنده  $j$  ام در صف مشتری است).

کمیته کردن زمان اتمام کل برنامه کاربردی به عنوان تنها معیار تخصیص وظایف، باعث می‌شود بیش تر وظیفه‌ها به دستگاه‌های سیار سریع تر (قدرت پردازش بالاتر) انتساب داده شوند. این کار منجر به مصرف سریع انرژی این گره‌ها شده و باتری آن‌ها زودتر از گره‌های دیگر تمام می‌شود. بنابراین، در این مقاله، علاوه بر زمان، انرژی باقی مانده گره‌ها به عنوان معیار دیگر بهینه‌سازی در تخصیص وظایف در نظر گرفته می‌شود. در یک تخصیص وظایف عادلانه، وظیفه‌ها باید به گونه‌ای توزیع شوند که طول عمر شبکه بیشینه شود. از این رو، از مدل انرژی ارائه شده در مقاله [۸] که از مدل انرژی مطرح شده در مسیریابی آگاه به انرژی شبکه‌های اقتضایی [۱۰] الهام گرفته است، استفاده می‌گردد. با استفاده از این مدل، به هر گره به نسبت انرژی اولیه آن، وظیفه تخصیص داده می‌شود. در واقع با استفاده از مدل انرژی ارائه شده در رابطه (۲)، انرژی مصرفی کل شبکه کاهش یافته و در عین حال از دادن وظیفه به گره‌هایی که انرژی کمی دارند اجتناب می‌گردد [۸]، یعنی سعی می‌کند ضمن کاهش انرژی مصرفی کل شبکه، حداقل انرژی باقی مانده گره‌ها و طول عمر آن‌ها را افزایش دهد [۱۴].

همچنین  $m$  دستگاه سیار، به کمک پروتکل شناسایی، در اطراف دستگاه سیار مشتری شناسایی شده‌اند. دستگاه سیار مشتری را با  $R_1$  و گره سرویس دهنده  $z$  ام با  $R_j$  نشان داده می‌شوند (در مجموع  $m+1$  گره برای تخصیص هریک از وظیفه‌ها، موجود است).

در مرحله جمع‌آوری اطلاعات زمینه‌ای، مدت زمان و انرژی مصرفی حاصل از اجرای یک وظیفه روی هریک از گره‌های سرویس دهنده تخمین زده می‌شود (در بخش ۳-۲ چگونگی تخمین این اطلاعات توضیح داده می‌شود). اطلاعات متغیرهای مسئله بهینه‌سازی در جدول ۱ نمایش داده شده است.

یک راه حل برای این مسئله، بردار تخصیص وظایف  $m+1$  عنصری  $b$  است که درایه  $b_j$  تعداد زیر برنامه‌های انتساب داده شده به گره  $z$  ام را مشخص می‌کند ( $b_j$  تعداد وظیفه‌هایی که به صورت محلی روی دستگاه سیار مشتری اجرا می‌شوند می‌باشد).

جدول ۱: نماد متغیرهای مسئله تخصیص وظایف در ابر سیار

نام متغیر	توضیحات
$n$	تعداد زیر برنامه‌های (وظیفه‌های) برنامه کاربردی
$m$	تعداد فراهم کنندگان سرویس
$R_2$	دستگاه سیار مشتری
$R_j$	فراهم کننده سرویس $z$ ام
$t_j$	مدت زمان اجرای یک وظیفه روی گره $R_j$
$e_j$	انرژی مصرفی اجرای یک وظیفه روی گره $R_j$ در واحد زمان
$et_j$	انرژی لازم برای انتقال واحد داده از گره مشتری به گره $R_j$
$E0_j$	انرژی اولیه گره $R_j$
$\alpha_j$	آستانه انرژی باقی مانده گره $R_j$ برای حاضر به همکاری بودن در واگذاری
$Vin$	اندازه داده ورودی یک وظیفه
$Vout$	اندازه داده خروجی یک وظیفه
$BW_j$	پهنای باند شبکه در کانال ارتباطی با گره $R_j$
$T_0$	مهلت زمانی

از آنجایی که وظیفه‌ها موازی و مستقل از هم روی دستگاه‌های سیار اجرا می‌شوند، همان طور که در [۱۳] اشاره شده، مدت زمان تکمیل اجرای کل برنامه برابر با بیشینه  $tl_j$ ها خواهد شد به طوری که  $tl_j$  زمان تکمیل اجرای وظیفه‌های انتسابی به گره  $z$  ام است. به علاوه، زمان اجرای الگوریتم تخصیص وظایف در دستگاه سیار مبدأ نیز مهم است و با متغیر ( $algoRuntime$ ) معرفی شده است. این زمان، با توجه به پیچیدگی محاسباتی الگوریتم تخصیص وظایف و میانگین چند بار اجرای آن، تخمین زده می‌شود. مدت زمان اجرای کامل برنامه به کمک رابطه (۱) محاسبه می‌شود. زمان تکمیل وظیفه‌ها برای گره مشتری ( $j=1$ ) برابر با مجموع زمان اجرای وظیفه‌هایی است که باید محلی انجام دهد. همچنین مدت زمانی در گره مشتری صرف ارسال داده‌ها به گره‌های سرویس دهنده می‌شود.

$$completeTime = \left( \max_{j \in \{1, \dots, m+1\}} tl_j \right) + algoRuntime$$

می شود. پردازش لازم برای عکس‌ها یکسان بوده و حجم هر عکس  $(Vin)$  ۲/۵۴ مگابایت است. دستگاه سیار کاربر طی پروتکل شناسایی، یک تلفن همراه هوشمند ( $R_2$ ) و یک لپ‌تاپ ( $R_3$ ) را پیرامون خود پیدا می کند ( $m=2$ ). در مرحله جمع‌آوری اطلاعات زمینه‌ای، پارامترهای مسئله به صورت زیر تعیین می شوند.

مقادیر بردارهای  $t$  و  $e$  یعنی زمان اجرا (برحسب ثانیه) و انرژی مصرفی (برحسب ژول) ناشی از اجرای یک وظیفه روی هر یک از سه دستگاه سیار به صورت زیر مشخص شده است:

$$t = \{24.4, 24.4, 5.2\} (s)$$

$$e = \{0.6147, 0.6147, 14.3\} (J)$$

برای سادگی، مقدار انرژی اولیه ( $E_{0j}$ ) هر سه وسیله یکسان و برابر ۱۶۳۵۰ (J) (انرژی اولیه دستگاه سیار کاربر) در نظر گرفته شده است. انرژی انتقال داده و حداقل انرژی لازم برای همکاری هر گره فراهم کننده سرویس نیز به صورت زیر مشخص شده است:

$$et = \{0.247, 1, 1\} (J)$$

$$\alpha = \{16000, 14000, 10000\} (J)$$

سایر پارامترهای مسئله به صورت زیر مقدار دهی می شوند:

$$BW = 1.867 (Mbps), T_0 = 100 (s), Vout = 1 (KB).$$

فرض کنید  $R_2$  و  $R_3$  هر یک، برنامه را روی دو عکس اجرا می کنند و یک عکس هم به صورت محلی روی دستگاه سیار کاربر بررسی می شود. بنابراین بردار تخصیص وظایف  $b$  به صورت زیر خواهد بود:

$$b = \{1, 2, 2\}$$

بدین ترتیب، زمان تکمیل وظیفه‌های هر گره ( $tl_j$ ) به صورت زیر به

دست می آید:

$$tl_1 = 1 * 24.4 + 5.46 + 1 = 29.86 (s)$$

که در آن  $sendDelay$  به صورت زیر محاسبه می شود:

$$sendDelay = (5-1) \frac{2.54}{1.867} = 5.46 (s) *$$

همان طور که در بخش ۳-۱، در مورد زمان بند ارسال فایل‌ها، توضیح داده خواهد شد،  $Qdelay$  برای اولین گره سرویس دهنده صفر است، بنابراین زمان تکمیل گره دوم برابر خواهد بود با:

$$tl_2 = 2 * (24.4 + \frac{2.54}{1.867}) + 0 = 51.53 (s) \quad (Qdelay = 0)$$

و زمان تکمیل اجرای گره سوم به صورت زیر محاسبه می شود:

$$tl_3 = 2 * (5.2 + \frac{2.54}{1.867}) + 2.73 = 15.86 (s)$$

که در آن زمان انتظار در صف پروکسی گره مشتری ( $Qdelay$ ) به صورت زیر به دست می آید:

$$Qdelay = 2 * \frac{2.54}{1.867} = 2.73 (s)$$

با در نظر گرفتن زمان اجرای الگوریتم برابر یک ثانیه، زمان اتمام کل برنامه کاربردی برابر با بیشترین مقدار  $tl_j$  به علاوه یک، یعنی ۵۲/۵۳ خواهد بود. همچنین انرژی مصرفی هر یک از گره‌ها با توجه به وظیفه‌هایی که انجام داده اند به صورت زیر خواهد بود:

$$consumedE_1 = 1 \times (15) + 4 \times (0.247 \times 2.54) + 1 = 18.5 (J)$$

$$consumedE_2 = 2 \times (24.4 \times 0.6147) = 30 (J)$$

$$consumedE_3 = 2 \times (5.2 \times 14.3) = 148.8 (J)$$

$$energy = \sum_{j=1}^{m+1} \frac{consumedE_j}{E_{0j} - consumedE_j}, \quad (2)$$

$$consumedE_j = \begin{cases} b_j * (t_j * e_j) + \sum_{r=2}^{m+1} b_r * (e_r * Vin) + \\ \quad algoEnergy, j = 1 \\ b_j * [(t_j * e_j) + (e_j * Vout)], j > 1 \end{cases}$$

در رابطه (۲)، انرژی مصرفی هر گره با  $consumedE_j$  و انرژی اولیه آن با  $E_{0j}$  نشان داده شده است. انرژی مصرفی گره مشتری ( $j=1$ )، شامل انرژی مصرفی برای اجرا وظیفه‌های محلی، اجرای الگوریتم تخصیص وظایف، و انرژی لازم برای ارسال داده‌های مربوط به وظیفه‌های واگذار شده می باشد. انرژی مصرفی سرویس دهنده ( $j>1$ )، شامل انرژی مصرفی برای اجرای وظیفه‌های انتساب داده شده آن به علاوه انرژی مصرفی برای ارسال نتایج به گره مشتری است. بنابراین، بهینه‌سازی دومعیاره تخصیص وظایف برای یک درخواست واگذاری در ابر سیار، به صورت زیر فرموله خواهد شد:

$$\text{Minimize } complete \text{ time} \quad (3)$$

$$\text{Minimize } energy \quad (4)$$

$$\text{Subject to } completeTime \leq T_0, \quad (5)$$

$$E_{0j} - consumedE_j \geq \alpha_j \quad \text{for all } j = 1 \dots m + 1, \quad (6)$$

$$\sum_{j=1}^{m+1} b_j = n. \quad (7)$$

قیود ۵ تا ۷ امکان‌پذیر بودن جواب‌های مسئله را بررسی کرده و نیازمندی‌های کیفیت سرویس را تأمین می کنند.

- شرط مهلت زمانی اجرا (۵): این شرط که جزء شروط کیفیت سرویس است، ضمانت می کند زمان اجرای برنامه کاربردی، از مهلت زمانی تعیین شده توسط کاربر ( $T_0$ ) بیش تر نباشد.
- سیاست پذیرش فراهم کنندگان سرویس (۶): هر یک از گره‌های سرویس دهنده، در صورت تضمین آستانه انرژی باقی مانده ( $\alpha_j$ ) حاضر به همکاری در واگذاری هستند که این آستانه را ضمن پروتکل شناسایی اعلام می کنند.
- شرط (۷) اطمینان حاصل می کند، تمام زیر برنامه‌های برنامه‌ی کاربردی به گره‌ای تخصیص داده شده باشند، چه انتساب به فراهم کنندگان سرویس (اجرای راه دور) باشد و چه به دستگاه سیار مشتری (اجرای محلی).

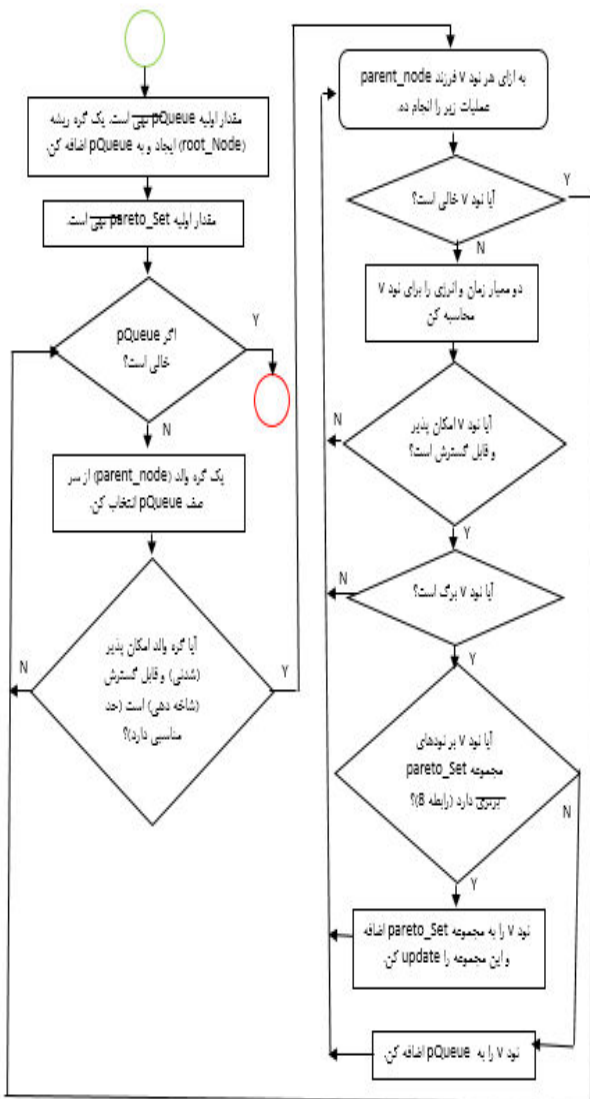
در ادامه به منظور نشان دادن طرز عملکرد مدل مسئله‌ی تخصیص وظایف و ملموس تر شدن مدل، یک مثال عددی برگرفته شده از اجرای یک برنامه کاربردی بیان می شود.

سناریوی زیر را در نظر بگیرید. دستگاه سیار کاربر ( $R_1$ ) می خواهد یک نرم افزار تشخیص چهره را روی پنج عکس اجرا کند. در اینجا، اجرای نرم افزار بر روی یک عکس به عنوان یک وظیفه در نظر گرفته

توابع هدف  $X$  ( $f_i(X)$ ) با توابع هدف  $S$  ( $f_i(S)$ ) مقایسه و شروط زیر (پارتو نبودن  $X$ ) بررسی گردد:

$$\begin{aligned} \exists S \in PS : \\ \forall i=1, \dots, m : f_i(X) \geq f_i(S) \\ \exists j : f_j(X) > f_j(S) \end{aligned} \quad (8)$$

اگر حتی یکی از نقاط پارتو بر جواب فعلی چیره شد، آن جواب باید حذف شود. در روش شاخه و حد می توان به جای مقدار توابع هدف ( $f_i(X)$ ) از حد پایین آن توابع ( $L_i(X)$ ) استفاده نمود [۱۷]. در این مقاله، یک الگوریتم شاخه و حد چندهدفه با عنوان MRABB، طراحی شده است که هدف آن یافتن مجموعه جواب‌هایی برای تخصیص وظایف در واگذاری وظیفه‌ها به ابر سیار است که دو معیار زمان اجرا و انرژی مصرفی را همزمان کمینه سازد. فلوجارت الگوریتم MRABB در شکل ۲ نشان داده شده است.



شکل ۲: فلوجارت الگوریتم پیشنهادی MRABB

جزئیات بیشتر الگوریتم MRABB در شبه‌کد جدول ۲ بیان شده است. در این الگوریتم، تابع  $feasible$  قیود تعریف شده در روابط (۵) تا (۷) را بررسی کرده و اگر جواب فعلی شدنی بود، حد آن را تخمین

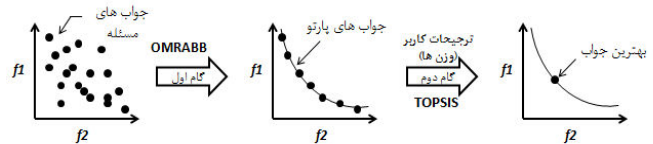
مجموع نسبت انرژی مصرفی هر گره به انرژی باقی مانده آن، طبق رابطه (۲)، به صورت زیر محاسبه می‌شود:

$$energy = \frac{18.5}{16331.5} + \frac{30}{16320} + \frac{148.8}{16201.2} = 0.01215$$

همچنین بردار تخصیص وظایف فوق تمام قیود مسئله را رعایت می‌کند.

## ۲-۱-۱-۲ حل مسئله تخصیص وظایف چندهدفه در ابر سیار

مسئله تخصیص وظایف فوق مانند مسئله کوله‌پشتی ۰-۱ چندهدفه، ان‌پی-سخت می‌باشد [۱۵، ۱۶]. برای حل این مسئله بهینه‌سازی چندهدفه در این مقاله، ابتدا الگوریتمی مبتنی بر شاخه و حد بنام MRABB برای به دست آوردن مجموعه جواب‌های بهینه پارتو ارائه خواهد شد، سپس با استفاده از بردار ترجیحات کاربر و روش Topsis، جواب بهینه نهایی تعیین می‌شود، فرایند دومرحله‌ای تخصیص وظایف در روش پیشنهادی، در شکل ۱ نشان داده شده است.



شکل ۱: فرایند دومرحله‌ای تخصیص وظایف در ابر سیار

## ۲-۱-۲ الگوریتم MRABB

در اکثر الگوریتم‌های بهینه‌سازی، با توجه به پیچیده بودن مسئله (غیر خطی بودن توابع هدف، داشتن قیود و گسسته بودن متغیر مسئله)، از ایده شاخه و حد برای یافتن بهینه مطلق در زمان مناسب استفاده می‌شود [۱۶، ۱۷]. الگوریتم شاخه و حد از جمله روش‌هایی است که به صورت گسترده و کارا در حل بهینه‌سازی چندهدفه استفاده می‌شود. این الگوریتم ساده و قابل فهم بوده و جواب بهینه مطلق را به دست می‌دهد [۱۶، ۱۸].

در الگوریتم شاخه و حد، درخت فضای حالت به صورت کامل جستجو و هرس می‌شود. در بررسی هر گره از درخت، حد پایین جواب (در کمینه‌سازی) تخمین زده می‌شود و با مقدار کمینه تاکنون مقایسه می‌گردد. در نسخه چندهدفه شاخه و حد، یک مجموعه نقاط پارتو نظر گرفته می‌شود که نقاط نزدیک به بهینه پارتو که تاکنون یافت شده‌اند را نگهداری می‌کند. مقدار اولیه این مجموعه، تهی است و اولین جواب شدنی که پیدا شود، در این مجموعه قرار می‌گیرد. ضمن جستجو، اگر جوابی از مسئله پیدا شود که بهتر از نقاط پارتو فعلی باشد، نقاط پارتویی که جواب یافته شده بر آن‌ها چیره شده است از مجموعه پارتو حذف شده و جواب فعلی به مجموعه اضافه می‌گردد. به این ترتیب، مجموعه نقاط پارتو به روزرسانی شده تا همیشه نقاط بهینه پارتو را نگهداری نماید.

فرض کنید  $PS$  مجموعه نقاط پارتو فعلی و  $X$  جواب شدنی فعلی مسئله باشند. به ازای هر نقطه  $S$  از مجموعه پارتو، لازم است مقادیر

این الگوریتم الزاماً میانگین انرژی مصرفی را کمینه نمی کند. در نتیجه، حد پایین انرژی همیشه کم تر یا مساوی مقدار واقعی انرژی نخواهد بود. البته این الگوریتم تخمین خوبی از بهینه ایده آل انرژی فراهم می کند و به همین دلیل، الگوریتم MRABB با دقت خوبی مجموعه جواب نزدیک به بهینه را به دست می آورد.

خطوط ۴ تا ۱۶، چیرگی جواب فعلی بر مجموعه جواب های پارتو را طبق رابطه (۸) بررسی می کند.

جدول ۳: تابع حد در الگوریتم شاخه و حد MRABB

1:	<b>Boolean bound</b> (curSt, paretoSet) //Input: curSt is the current state node in search tree and paretoSet is optimal Pareto up to now //Output : False if the node should be pruned, elsewhere True
2:	curSt.timeBound=timeLowerBound (curSt );
3:	curSt.energyBound =energyLowerBound(curSt );
4:	SdominatecurSt = False;
5:	<b>for</b> (i = 1 to paretoSet.size ) <b>do</b>
6:	NodeBIP paretoPoint = paretoSet.get(i);
7:	<b>if</b> (curSt.timeBound >= paretoPoint.time &&
8:	curSt.energyBound >= paretoPoint.energy)
9:	<b>if</b> (curSt.timeBound > paretoPoint.time
10:	curSt.energyBound > paretoPoint.energy)
11:	SdominatecurSt = True;
12:	<b>break</b> ;
13:	<b>if</b> (SdominateCurSt == True)
14:	<b>return</b> False; // Prune the curSt
15:	<b>else</b>
16:	<b>return</b> True; // Branch the curSt

پیچیدگی الگوریتم MRABB ارائه شده  $O(n^m)$  است که در آن،  $n$  تعداد وظیفه های برنامه کاربردی و  $m$  تعداد گره های مجاور است. اما با توجه به اینکه تعداد گره های موجود در همسایگی یک گامی معمولاً کم است، ابر سیار متشکل از موبایل های پیرامون با سایز بزرگ معمولاً تشکیل نمی شود، به همین دلیل پیچیدگی زمانی الگوریتم پیشنهادی محدود خواهد بود. از طرفی به دلیل هرس کردن شاخه های غیرامیدبخش، زمان اجرای الگوریتم MRABB نسبت به جستجوی تمام درخت فضای حالت بسیار کارتر خواهد بود.

### ۱-۲-۳- تصمیم گیری چندشاخصه

با استفاده از الگوریتم MRABB مجموعه جواب های بهینه ی پارتو به دست می آیند. اما برای اجرای واگذاری، به یک جواب نهایی برای تخصیص وظایف نیاز است. برای این منظور، به کمک روش های تصمیم گیری چندشاخصه، جواب نهایی از میان مجموعه جواب های بهینه پارتو به دست آمده، تعیین می شود [۱۹]. ماتریس تصمیم گیری برای مسئله تصمیم گیری چندشاخصه، یک ماتریس  $2 \times R$  بعدی بنام  $R$  خواهد بود که هر سطر آن متناظر با هریک از گزینه ها (هریک از جواب های مجموعه پارتو) و ستون اول و دوم نشان دهنده دو معیار تصمیم گیری یعنی زمان و انرژی خواهند بود.  $R_{ij}$  مقدار تابع هدف نام برای گزینه  $i$ ام را نشان می دهد.

تابع  $bound$ ، حد پایین زمان و انرژی را تخمین زده و با مجموعه نقاط پارتو یافته شده، مقایسه می کند. در خطوط ۱۲ و ۱۳ از کد، زمان اجرا و انرژی طبق رابطه های (۱) و (۲) محاسبه می شود. هریک از جواب های شدنی، با نقاط مجموعه پارتو مقایسه می شوند. اگر حد پایین جواب فعلی توسط هیچیک از نقاط مجموعه پارتو چیره نشد، جواب فعلی به مجموعه جواب پارتو اضافه می شود. همچنین مجموعه نقاط پارتو به روزرسانی می شود و اگر جواب پیدا شده از هریک جواب های پارتو تاکنون یافت شده، بهتر بود، آن نقطه را حذف می کند (خطوط ۱۶ تا ۱۸).

جدول ۲: شبه کد الگوریتم پیشنهادی MRABB

1:	<b>MRABB</b> (n, m, profiles) //Input: n and m are number of subtasks and mobile devices, respectively. Profiles //represent task, devices and network information. //Output: Optimal Pareto Solution Set of the problem
2:	pQueue = initPriorityQueue();
3:	root = makeRootNode();
4:	pQueue.add(root);
5:	paretoSet = $\emptyset$ ;
6:	<b>while</b> (!pQueue.empty())
7:	parentState = pQueue.poll();
8:	<b>if</b> (feasible(parentState) && bound(parentState, paretoSet) )
	// parentState may not satisfy feasible constraints or may not have better bound than current solution in Pareto
9:	remainSubTsk = parentState.remainSubtasks();
10:	<b>for</b> (t = 1 to remainSubTsk) <b>do</b>
11:	curState = initState(parentState, t);
12:	time = completeTime(curState, profiles);
13:	energy = residual_Energy(curState, profiles);
14:	curState.setCriteria(time, energy);
15:	<b>if</b> (feasible (curState) && bound (curState, paretoSet) )
16:	<b>if</b> ( remTsk + t == n && paretoSet.NonDominate(curState) )
17:	paretoSet.addPareto(curState);
18:	updatePareto(paretoSet , curState);
19:	<b>else</b>
20:	pQueue.put(curState);
21:	<b>end while</b>
22:	<b>return</b> paretoSet;

جدول ۳ شبه کد تابع حد در الگوریتم MRABB را نشان می دهد. برای تخمین حد پایین زمان (خط ۲)، از روش حریصانه (همانند الگوریتم بهینه سازی زمان در [۸]) استفاده شده است. یعنی برای وظیفه  $k+1$ ام، گرهی  $z$ ام که با توجه به وظیفه های قبلی خود می تواند آن را در زمان کوتاه تری تمام نماید، انتخاب می شود و مدت زمان اشغال آن گره به روز می شود. در آخر، حد پایین زمان جواب بخشی فعلی، بیشینه زمان تخمینی اتمام وظیفه های هریک از گره ها خواهد بود. حد پایین انرژی نیز مانند حد زمان محاسبه می شود (خط ۳)، با این تفاوت که به جای انتخاب گره با کم ترین مدت زمان اجرا، گره با کم ترین مقدار تابع انرژی (مطابق رابطه ۲) برای هریک از وظیفه های باقی مانده انتخاب می گردد. همان طور که در مقاله [۸] ذکر شده است،

گام چهارم، محاسبه فاصله و میزان جدایی هر یک از گزینه‌ها است. برای محاسبه فاصله می‌توان از فاصله اقلیدسی  $n$  بعدی استفاده نمود. بنابراین فاصله گزینه نام از راه‌حل ایده‌آل مثبت ( $A^+$ ) و راه‌حل ایده‌آل منفی ( $A^-$ ) به صورت زیر محاسبه می‌شود:

$$d_i^+ = \left\{ \sum_{j=1}^q (v_{ij} - v_j^+)^2 \right\}^{1/2}, \quad i = 1, 2, \dots, r \quad (14)$$

$$d_i^- = \left\{ \sum_{j=1}^q (v_{ij} - v_j^-)^2 \right\}^{1/2}, \quad i = 1, 2, \dots, r$$

گام پنجم، محاسبه شباهت یا نزدیکی نسبی گزینه نام به راه‌حل ایده‌آل است. نزدیکی نسبی به صورت زیر تعریف می‌شود:

$$c_i = \frac{d_i^-}{d_i^+ + d_i^-}, \quad i = 1, 2, \dots, r \quad (15)$$

از آنجایی که  $d_i^+ \geq 0$  و  $d_i^- \geq 0$  است، بنابراین  $c_i \in (0, 1)$  است. هر چه  $A_i$  به  $A^+$  نزدیک‌تر و از  $A^-$  دورتر باشد،  $c_i$  به یک نزدیک‌تر خواهد شد.

گام ششم، رتبه‌بندی گزینه‌ها است. می‌توان گزینه‌های موجود در مسئله مفروض را بر اساس ترتیب نزولی  $c_i$  رتبه‌بندی نمود. در آخر جواب با بیش‌ترین مقدار  $c_i$  به عنوان جواب نهایی تعیین می‌گردد.

### ۳- نرم‌افزار OMC

در این مقاله، به منظور فراهم نمودن امکان واگذاری برنامه‌های کاربردی موبایل، نرم‌افزار واگذاری در ابر سیار (به اختصار OMC) طراحی و پیاده‌سازی شده است که مراحل واگذاری را مدیریت می‌کند.

#### ۳-۱- معماری سیستم OMC

نرم‌افزار OMC در دو بخش طراحی شده است: سمت مشتری و سمت سرویس‌دهنده. شکل ۳ معماری کلی نرم‌افزار پیشنهادی را نشان می‌دهد. هر دستگاه سیار، هر دو بخش نرم‌افزار را دارا بوده و با توجه به این که در هر واگذاری، نقش مشتری یا فراهم‌کننده سرویس را ایفا کند، بخش مربوطه را به اجرا درمی‌آورد.

سمت مشتری OMC از چهار مؤلفه اصلی تشکیل شده است: (۱) مدیریت اطلاعات زمینه‌ای<sup>۱۲</sup>، (۲) تخصیص وظایف، (۳) پروکسی سمت مشتری و (۴) مدیریت واگذاری.

(۱) مدیریت اطلاعات زمینه‌ای: این مؤلفه اطلاعات مورد نیاز درباره دستگاه‌های سیار مجاور، برنامه کاربردی و شبکه را به کمک جمع‌آوری‌کننده هریک به دست می‌آورد. پروفایل دستگاه‌های سیار شامل اطلاعاتی درباره قدرت پردازش، مدل مصرف انرژی و انرژی باقی‌مانده باتری دستگاه سیار است. پروفایل برنامه کاربردی، حاوی اطلاعات برنامه مانند زیربرنامه‌های قابل واگذاری، زمان اجرا و اندازه ورودی/خروجی هر زیربرنامه است.

مدیریت اطلاعات زمینه‌ای پس از اتمام واگذاری، اطلاعات اجرایی (زمان و انرژی مصرفی) وظیفه‌های اجرا شده در واگذاری اخیر را جمع‌آوری کرده و در پایگاه داده اطلاعات زمینه‌ای ذخیره می‌نماید.

$$R = \begin{matrix} A_1 & A_2 \\ \left[ \begin{array}{cc} R_{11} & R_{12} \\ R_{21} & R_{22} \\ \dots & \dots \\ R_{r1} & R_{r2} \end{array} \right] & \begin{array}{l} \text{Solution 1} \\ \text{Solution 2} \\ \dots \\ \text{Solution r} \end{array} \end{matrix} \quad (9)$$

به‌طور کلی روش‌های مختلفی برای تصمیم‌گیری چندشاخصه وجود دارند [۱۲، ۱۹]. در این مقاله، از روش TOPSIS [۲۰] برای تعیین جواب نهایی مصالحه‌گر استفاده شده است. دلایل انتخاب روش TOPSIS عبارت‌اند از: (۱) رویه محاسباتی سریع و سبکی دارد، (۲) معیارها می‌توانند وابسته یا مستقل باشند، (۳) توانایی اعمال ضرایب وزنی روی معیارهای مختلف را داراست [۱۲، ۱۹]. روش TOPSIS مبتنی بر مفهوم فاصله از جواب‌های ایده‌آل مثبت و منفی است. بدان معنی که گزینه‌ی انتخابی باید دارای کم‌ترین فاصله از جواب ایده‌آل مثبت بوده و در عین حال دارای دورترین فاصله از راه‌حل ایده‌آل منفی باشد [۲۰]. منظور از جواب ایده‌آل مثبت و منفی به ترتیب جواب با بهترین و بدترین مقادیر معیارها است. فرایند TOPSIS شامل مراحل زیر است [۱۲]:

گام اول، تبدیل ماتریس تصمیم‌گیری به ماتریس نرمال و بی‌مقیاس شده است. برای نرمال‌سازی مقادیر هریک از توابع هدف (هر ویژگی)، روش‌های متعددی وجود دارد، در اینجا طبق رابطه زیر، مقادیر هر ستون از ماتریس تصمیم‌گیری  $R$  نرمال می‌شود:

$$p_{ij} = \frac{R_{ij}}{\sum_{v=1}^r R_{vj}}, \quad i = 1, 2, \dots, r; j = 1, 2, \dots, q \quad (10)$$

بنابراین ماتریس تصمیم‌گیری نرمال شده به صورت زیر خواهد شد:

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \\ \dots & \dots \\ p_{r1} & p_{r2} \end{bmatrix} \quad (11)$$

گام دوم، ایجاد ماتریس بی‌مقیاس وزین با مفروض وجود بردار  $W$  است. در روش تخصیص وظایف پیشنهادی، میزان اهمیت (وزن) هریک از دو معیار زمان و انرژی از کاربر گرفته می‌شود. بنابراین هر درایه از ماتریس بی‌مقیاس وزین به صورت زیر محاسبه می‌شود:

$$v_{ij} = w_j * p_{ij}, \quad i = 1, 2, \dots, r; j = 1, 2, \dots, q \quad (12)$$

گام سوم، تعیین راه‌حل ایده‌آل مثبت و راه‌حل ایده‌آل منفی است. در واقع برای رتبه‌بندی و تعیین جواب نهایی لازم است گزینه‌ها با راه‌حل‌های ایده‌آل مثبت و منفی مقایسه شوند.

$$A^+ = \left\{ \left( \max_i v_{ij} \mid j \in J \right), \left( \min_i v_{ij} \mid j \in J' \right) \mid i = 1, \dots, r \right\}$$

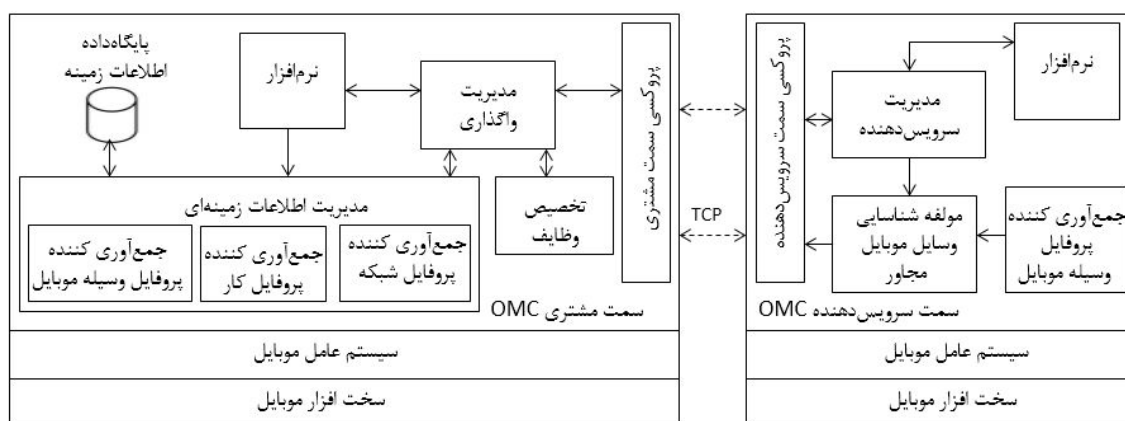
$$= \{v_1^+, v_2^+, \dots, v_q^+\} \quad (13)$$

$$A^- = \left\{ \left( \min_i v_{ij} \mid j \in J \right), \left( \max_i v_{ij} \mid j \in J' \right) \mid i = 1, \dots, r \right\}$$

$$= \{v_1^-, v_2^-, \dots, v_q^-\}$$

که در آن  $J$  و  $J'$  به ترتیب معیارهای سود و هزینه هستند.

قطع شود، OMC مانند [۵] اجرای وظیفه‌های ارسالی به آن دستگاه را به صورت محلی از سر می‌گیرد. همچنین، این مؤلفه زمان‌بندی ارسال فایل‌های لازم هر وظیفه به گره سرویس‌دهنده آن را بر عهده دارد. در این زمان‌بندی، تمام فایل‌های مربوط به یک دستگاه به صورت متوالی برای آن ارسال می‌شوند و الویت انتخاب دستگاه‌ها به ترتیب صعودی سرعت پردازش آن‌هاست. در نتیجه، گره‌هایی که قدرت پردازش کم‌تری دارند، زمان بیش‌تری برای انجام وظیفه‌های خود خواهند داشت.



شکل ۳: دید کلی از معماری پیشنهادی OMC

(۱) شناسایی دستگاه‌های سیار پیرامون: در ابتدا دستگاه سیار مشتری با استفاده از پروتکل شناسایی، دستگاه‌های سیار مجاورش را پیدا می‌کند. در پروتکل شناسایی، دستگاه‌های سیار حاضر به همکاری، آمادگی خود (به همراه پروفایل خود) را در پیامی به صورت همگانی و دوره‌ای ارسال می‌کنند.

(۲) جمع‌آوری اطلاعات زمینه: سمت مشتری در OMC، به کمک مدیریت اطلاعات زمینه‌ای، سایر اطلاعات موردنیاز درباره برنامه‌ی کاربردی، دستگاه‌های سیار و شبکه را جمع‌آوری کرده، در اختیار مؤلفه تخصیص وظایف برای تصمیم‌گیری قرار می‌دهد.

(۳) تخصیص وظایف: مؤلفه تخصیص وظایف از اطلاعات زمینه‌ای به دست‌آمده برای مقداردهی پارامترهای مسئله بهینه‌سازی چندهدفه تخصیص وظایف در ابر سیار و حل آن با روش پیشنهادی، استفاده می‌کند. همان‌طور که بیان شد، تخصیص وظایف پیشنهادی شامل دو مرحله به دست آوردن مجموعه جواب‌های پارتو (MRABB) و تعیین جواب بهینه نهایی تخصیص وظایف به کمک روش‌های تصمیم‌گیری چندشاخصه است. سپس، در مؤلفه پروکسی سمت مشتری در OMC، با استفاده از زمان‌بندی انتقال فایل‌ها و طبق ماتریس تخصیص وظایف به دست‌آمده، وظیفه‌هایی که قرار است واگذار شوند به همراه فایل ورودی آن‌ها بین گره‌های فراهم‌کننده مربوطه توزیع می‌شوند.

پس از اتمام هر واگذاری، داده‌های این پایگاه داده به‌روز می‌شوند. OMC از این اطلاعات تاریخچه‌ای جهت یادگیری و بهبود تخمین زمان و انرژی مصرفی وظیفه‌ها در واگذاری‌های آینده استفاده می‌کند.

(۲) تخصیص وظایف: مؤلفه تخصیص وظایف از اطلاعات زمینه‌ای به دست‌آمده به عنوان پارامترهای مسئله بهینه‌سازی (روابط ۷-۱) استفاده کرده و تعیین می‌کند هر وظیفه توسط کدام دستگاه سیار اجرا شود.

(۳) پروکسی در سمت مشتری: این مؤلفه، ارتباطات شبکه‌ای بین دستگاه سیار مشتری و فراهم‌کنندگان سرویس را نظارت و مدیریت می‌کند. در صورتی که ارتباط یکی از سرویس‌دهنده‌ها با دستگاه مشتری

(۴) مؤلفه مدیریت واگذاری: این مؤلفه، ارتباطات و هماهنگی‌های لازم بین بخش‌های سمت مشتری با یکدیگر و همچنین ارتباط آن‌ها با برنامه‌ی کاربردی را هماهنگ کرده و کل فرایند واگذاری را مدیریت می‌نماید.

در سمت سرویس‌دهنده OMC نیز چهار مؤلفه اصلی وجود دارد: (۱) مؤلفه شناسایی، (۲) پروفایلی دستگاه سیار، (۳) پروکسی سمت سرویس‌دهنده و (۴) مدیریت فراهم‌کننده سرویس واگذاری. هر دستگاه سیار که مایل به همکاری در واگذاری باشد، آمادگی خود را از طریق پروتکل شناسایی پیش‌فعال ارائه‌شده در [۲۱] به همه اعلام می‌نماید. در این پروتکل، هر دستگاه به صورت دوره‌ای پیامی حاوی اطلاعات پروفایل خود را به صورت همگانی ارسال می‌کند. پروکسی سمت سرویس‌دهنده ارتباطات میان فراهم‌کننده سرویس و مشتری را مدیریت می‌کند. مؤلفه مدیریت فراهم‌کننده سرویس واگذاری، هماهنگی‌های لازم بین مؤلفه‌های سمت سرویس‌دهنده OMC و همچنین ارتباط آن‌ها با برنامه کاربردی را برای اجرای زیربرنامه‌های انتساب داده شده به آن دستگاه سیار، مدیریت می‌کند.

### ۳-۲- فرایند واگذاری در OMC

هنگامی که سمت مشتری در OMC، یک درخواست واگذاری برای یک برنامه‌ی کاربردی دریافت می‌کند، در صورتی که آن برنامه قابل واگذاری باشد، سمت مشتری در OMC فرایند واگذاری را به صورت زیر آغاز می‌کند:

ارتباط با شبکه، اجرای وظایف و واسط کاربری هر نسخه از نرم افزار در یک ریسمان مجزا اجرا می شوند که با بهره گیری از روش چندریسمانی، همزمانی مناسبی میان آن ها ایجاد شده است.

مجموعه ای از چهار دستگاه سیار که مشخصات آن ها در جدول ۴ نشان داده شده، در آزمایش ها استفاده شده است. دستگاه های سیار از طریق شبکه WiFi با متوسط پهنای باند واقعی اندازه گیری شده  $1/86$  Mbps در یک شبکه اقتصایی، با یکدیگر در ارتباط هستند. مهلت زمان انقضا برای برنامه ی کاربردی ( $T_0$ ) ۱۰۰۰ ثانیه می باشد.

برای اندازه گیری میزان انرژی مصرفی در تلفن های هوشمند (A) و (B) از نرم افزار PowerFour [۲۵] و در لپ تاپ ها (C) و (D) از نرم افزار Joumeter [۲۶] استفاده شده است. انرژی محاسبه شده شامل انرژی مصرفی CPU و انرژی مصرفی برای انتقال داده ها می باشد.

برای نشان دادن میزان بهبود در کارایی و ذخیره سازی انرژی روش پیشنهادی و همچنین توانایی آن در مدیریت مصالحه میان دو معیار زمان و انرژی، الگوریتم پیشنهادی MRABB با دو الگوریتم بهینه ساز زمان و آگاه از انرژی ارائه شده در سردیپیتی [۸] مقایسه شده است. در الگوریتم بهینه ساز زمان، به صورت تناوبی، برای هر وظیفه، گره مقصد با کم ترین زمان انجام شدن وظیفه انتخاب می شود. زمان اتمام یک وظیفه با در نظر گرفتن زمان اجرای آن وظیفه روی گره مقصد و تعداد وظیفه های تخصیص داده شده قبلی روی آن گره محاسبه می گردد. در الگوریتم آگاه از انرژی، برای هر وظیفه گره با کم ترین مقدار انرژی مطابق رابطه ۲ انتخاب می گردد به طوری که شرط مهلت زمانی برای کل برنامه کاربردی تأمین شود.

در بخش آزمایش ها، دو سناریو برای ارزیابی و مقایسه ی الگوریتم MRABB با دو روش بهینه ساز زمان و آگاه از انرژی در نظر گرفته شده است. در سناریو اول، تأثیر تعداد و قدرت پردازشی دستگاه های سیار مختلف به عنوان فراهم کنندگان سرویس بررسی می شود. در سناریو دوم، تأثیر افزایش تعداد وظیفه های یک برنامه کاربردی بر عملکرد سه الگوریتم تخصیص وظایف مورد ارزیابی قرار می گیرد. در این آزمایش ها، برای تعیین جواب نهایی سه حالت وزن دهی  $\{0/8, 0/2\}$ ،  $\{0/5, 0/5\}$  و  $\{0/8, 0/2\}$  که به ترتیب وزن را برای معیار زمان و انرژی (تعیین شده توسط کاربر) نشان می دهند، در نظر گرفته شده است. در هر بخش از آزمایش ها، سربار محاسباتی ناشی از اجرای الگوریتم تخصیص وظایف نیز بررسی و با روش های قبلی مقایسه خواهد شد.

در این آزمایش ها از چهار معیار زیر برای ارزیابی عملکرد الگوریتم تخصیص وظایف استفاده می شود:

- (۱) زمان کل واگذاری: شامل زمان اجرای الگوریتم تخصیص وظایف، اجرای وظایف و انتقال داده ها.
- (۲) انرژی: میانگین نسبت انرژی مصرفی به انرژی باقی مانده دستگاه های سیار، طبق فرمول ۲.

(۴) اجرای وظیفه ها: هر گره سرویس دهنده پس از دریافت فایل های ورودی وظیفه های خود، آن ها را اجرا کرده و نتایج آن را برمی گرداند. گره مشتری نیز وظیفه های که قرار است محلی انجام شوند را اجرا می کند.

(۵) جمع آوری و ادغام نتایج: در انتهای فرایند واگذاری، گره مشتری، نتایج جزئی را از فراهم کنندگان سرویس جمع آوری کرده و نتایج مرتبط را با هم ادغام می کند.

### ۳-۳- اطلاعات زمینه ای و پروفایل ها

برای تخمین زمان اجرا، اندازه خروجی، و انرژی مصرفی حاصل از اجرای هر زیربرنامه، روش های مختلفی پیشنهاد شده است [۵، ۶، ۲۲، ۲۳]. در این مقاله، با الهام از [۶]، از پروفایل دوگانه شامل پروفایل گره محور و وظیفه محور استفاده شده است. پروفایل گره محور، در صورت موجود بودن پروفایل اجرایی گره ها از واگذاری های قبلی مورد استفاده قرار می گیرد و اطلاعات دقیق تری فراهم می آورد.

پروفایل گره محور، یک پروفایل مبتنی بر تاریخچه حاصل از ۱۰ بار اجرای قبلی است که میانگین زمان های اجرا به عنوان تخمین زمان اجرا استفاده می شود. در صورت عدم وجود پروفایل گره محور (مثلاً وارد شدن گره جدید)، از پروفایل وظیفه محور استفاده می شود. در پروفایل وظیفه محور، از یک طرف، مدت زمان اجرای یک وظیفه روی یک ماشین مرجع با قدرت پردازشی واحد، به صورت برون خطی محاسبه می شود. از طرف دیگر، قدرت پردازشی هر دستگاه سیار با استفاده از محک کننده ی NBench [۲۴] و با توجه به مشخصات پردازشی آن وسیله، به صورت یک رتبه به دست می آید. بدین ترتیب، دستگاه مشتری با استفاده از قدرت پردازشی گره مقصد و بار کاری آن، قدرت پردازش فعلی آن دستگاه سیار را محاسبه می کند.

برای تخمین انرژی مصرفی از روش پیشنهادی در [۵] استفاده می شود که در آن برای هر دستگاه سیار، یک مدل مصرف انرژی ایجاد می شود. این مدل خطی انرژی، بر اساس تعداد دوره CPU مورد نیاز یک تابع، انرژی مصرفی حاصل از اجرای آن کد روی دستگاه سیار مذکور را پیش بینی می کند.

پروفایل شبکه (شامل پهنای باند و تأخیر شبکه) با استفاده از روش مطرح شده در [۵] اندازه گیری می شود. بدین صورت که یک بسته داده با اندازه ۱ KB به صورت دوره ای ارسال شده و زمان انتقال آن برای محاسبه توان عملیاتی شبکه مورد استفاده قرار می گیرد.

### ۴- ارزیابی

یک نمونه از نرم افزار OMC در دو نسخه برای اجرا روی سیستم عامل اندروید و ویندوز، شامل دو قسمت سمت مشتری و سمت سرویس دهنده، پیاده سازی شده است. همچنین از برنامه کاربردی تشخیص چهره به عنوان برنامه کاربردی استفاده شده است. این برنامه تعدادی عکس (به عنوان وظیفه های برنامه کاربردی) از ورودی گرفته و تمام چهره های موجود در هر عکس را شناسایی می کند. مؤلفه های

جدول ۴: مشخصات دستگاه‌های سیار استفاده شده در آزمایش‌ها

دستگاه سیار	CPU	حافظه	سیستم‌عامل	ظرفیت باتری (ژول)	شناسه
HTC wildfire S A510e	600 MHz ARMv6	۵۱۲ MB	Android 4.0.4	۱۶۳۸۳	A
HTC wildfire S A510e	600 MHz ARMv6	۵۱۲ MB	Android 4.0.4	۱۶۳۸۳	B
Sony VAIO VCP-CB17FX/B	2.4 GHz core i5-2430M	۶ GB	Windows 7	۱۴۱۸۰۴	C
DELL Vostro 1014	2.2 GHz core 2 duo	۴ GB	Windows 7	۱۳۷۵۴۱	D

این منظور، الگوریتم NSGA-II [۲۹] که یکی از الگوریتم‌های محبوب و پرکاربرد در زمینه بهینه‌سازی چندهدفه می‌باشد، با الگوریتم پیشنهادی MRABB مقایسه می‌گردد. الگوریتم NSGA-II مبتنی بر مرتب‌سازی ناچیره<sup>۱۳</sup> بوده و طی یکبار اجرا، مجموعه جواب پارتو را با همگرایی به مجموعه جواب‌های بهینه پارتو به دست می‌آورد. همچنین در این الگوریتم از روش فاصله تراکمی<sup>۱۴</sup> استفاده شده که در مجموعه جواب به دست آمده، پراکندگی و تنوع خوبی ایجاد کرده و از تراکم در جواب‌های پارتو جلوگیری کند.

#### ۴-۱- تأثیر تعداد و نوع دستگاه‌های سیار سرویس دهنده

در سناریوی اول، تأثیر تعداد و قدرت پردازشی دستگاه‌های سیار مختلف به‌عنوان فراهم‌کنندگان سرویس بررسی می‌شود. همچنین، عملکرد الگوریتم پیشنهادی MRABB در صورت ناهمگن بودن فراهم‌کنندگان سرویس علاوه بر تعداد آن‌ها مورد ارزیابی قرار می‌گیرد. گروه‌های مختلف دستگاه‌های سیار تشکیل‌دهنده ابر سیار، به‌صورت زیر در نظر گرفته شده است:

{A, B}, {A, D}, {A, C}, {A, B, D}, {A, B, C}, {A, C, D}, {A, B, C, D}.

در همه حالات این آزمایش، گره A به‌عنوان مشتری و بقیه گره‌ها نقش فراهم‌کننده سرویس را بازی می‌کنند. در این سناریو، از برنامه تشخیص چهره با ۱۰ عکس با اندازه ۳۸۰ KB به‌عنوان وظیفه‌های برنامه کاربردی برای واگذاری استفاده شده است.

این آزمایش نشان می‌دهد، با افزایش تعداد و همچنین قدرت پردازش دستگاه‌های سیار، زمان کل واگذاری کاهش و در صورت نبود گره‌ی با نرخ مصرف انرژی بالا (D)، انرژی مصرفی نیز کاهش می‌یابد. از مشاهده‌ی نمودارهای شکل ۴ نتایج زیر به دست می‌آید:

در حالت AB (گره‌های همگن)، مقادیر چهار نمودار در سه روش تخصیص وظایف، شبیه به هم است. در گروه‌های AD و ABD، گره‌ها ناهمگن می‌باشند و وقتی گره‌های ابر سیار ناهمگن باشند (قدرت پردازشی و نرخ مصرف انرژی دو گره A و B با گره‌های C و D تفاوت بسیاری دارد)، بین زمان و انرژی مصرفی دو الگوریتم بهینه‌ساز زمان آگاه از انرژی با توجه به نحوه تخصیص وظایف آن‌ها، اختلاف چشم‌گیری به وجود می‌آید و ضرورت مدیریت مصالحه میان زمان و انرژی بیش‌تر جلوه می‌کند. این اختلاف وقتی نرخ مصرف انرژی، عکس سرعت پردازش دستگاه باشد (مانند گره D)، بیش‌تر می‌گردد. در این حالات (ناهمگن بودن گره‌ها)، الگوریتم بهینه‌ساز زمان کم‌ترین زمان اجرا، بیش‌ترین انرژی مصرفی، کم‌ترین میزان عدالت (معیار Jain)

(۳) انرژی: میانگین نسبت انرژی مصرفی به انرژی باقی‌مانده دستگاه‌های سیار، طبق فرمول ۲.

(۴) معیار Jain: برای ارزیابی عدالت در میزان انرژی مصرفی هر یک از گره‌ها از معیار Jain [۲۷] استفاده می‌شود و به صورت زیر محاسبه می‌گردد:

$$Jain Index = \frac{(\sum_{j=1}^m x_j)^2}{m \cdot \sum_{j=1}^m x_j^2} \quad (16)$$

که در آن  $m$  تعداد گره‌های شرکت‌کننده در واگذاری و  $x_j$  انرژی مصرفی گره  $j$ ام است. مقدار معیار Jain عددی بین (۰ و ۱) است و هرچه بیش‌تر باشد، تخصیص وظایف عدالت بیش‌تری را تأمین نموده است.

(۵) کمینه انرژی باقی‌مانده گره‌ها (طول عمر شبکه). برای پرهیز از پیچیدگی و تأثیرات ناشی از ناهمگن بودن سطح انرژی باتری دستگاه‌های مختلف سیار، انرژی اولیه تمام دستگاه‌های سیار یکسان و برابر با ۱۶۳۸۳ ژول مقداردهی شده است. بدین ترتیب، در بررسی معیار Jain، هرچه انرژی مصرفی گره‌ها بیش‌تر شبیه به هم بوده و میزان انرژی باقی‌مانده آن‌ها به هم نزدیک‌تر باشد، عدالت انرژی باقی‌مانده بهتر تأمین می‌شود.

در شکل ۴ و ۵، نمودارهای الف، ب، ج و د، به ترتیب مقادیر چهار معیار زمان کل واگذاری، انرژی، معیار Jain و کمینه انرژی باقی‌مانده را نشان می‌دهند. هریک از آزمایش‌های زیر چند بار تکرار شده و میانگین نتایج گزارش شده است.

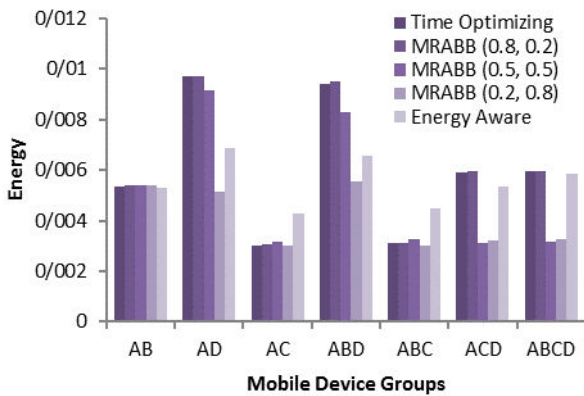
همچنین در آزمایش سوم، مصالحه میان دقت و زمان الگوریتم پیشنهادی MRABB نسبت به الگوریتم‌های تکاملی بهینه‌سازی چندهدفه، بررسی می‌گردد.

روش‌های متعددی برای حل مسائل بهینه‌سازی چندهدفه وجود دارد [۱۶، ۱۸، ۳۱-۲۸]. برخی روش‌های مانند الگوریتم پیشنهادی MRABB مبتنی بر شاخه و حد هستند و مجموعه جواب پارتو بهینه را به دست می‌آورند و برخی دیگر مبتنی بر الگوریتم‌های مکاشفه‌ای و تکاملی هستند. دسته دوم الگوریتم‌ها (الگوریتم‌های تکاملی)، اگرچه زمان اجرای کم‌تری نسبت به الگوریتم‌های مبتنی بر شاخه و حد دارند ولی مجموعه جواب بهینه مطلق را تضمین نمی‌کنند [۱۸]. همچنین سائز مجموعه جواب‌های پارتو حاصل از الگوریتم‌های تکاملی کم‌تر از مجموعه جواب‌های پارتو واقعی است، در نتیجه این احتمال وجود دارد که جواب نهایی بهینه از دست برود.

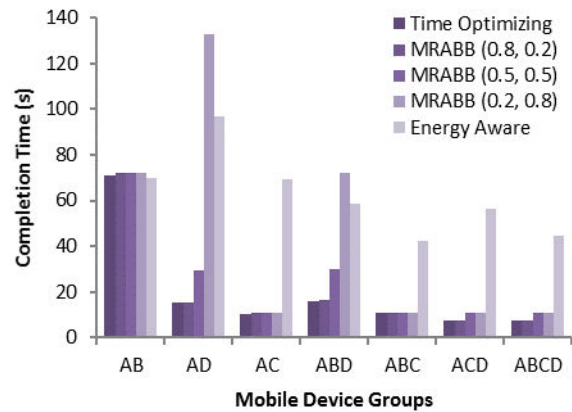
در این بخش از آزمایش‌ها، تفاوت این دو دسته الگوریتم از نظر دقت و تعداد مجموعه جواب‌های پارتو به دست آمده، بررسی می‌گردد. برای

الگوریتم پیشنهادی با اوزان  $\{0.8/0.2\}$  و  $\{0.5/0.5\}$  به طور خاص، در گروه‌های AD و ABD، توانسته انرژی مصرفی بهتری حتی نسبت به الگوریتم آگاه از انرژی به دست آورد. ولی چون برای کاهش میانگین مصرف انرژی، وظیفه کم‌تری به گره با نرخ مصرف انرژی بالا (D) داده است، معیار Jain و توازن کم‌تری در توزیع دارد. حتی در گروه‌های ACD و ABCD، میانگین انرژی مصرفی الگوریتم پیشنهادی از هر دو روش دیگر کم‌تر شده است (برخلاف دو روش دیگر، به گره D وظیفه‌ای تخصیص نمی‌دهد).

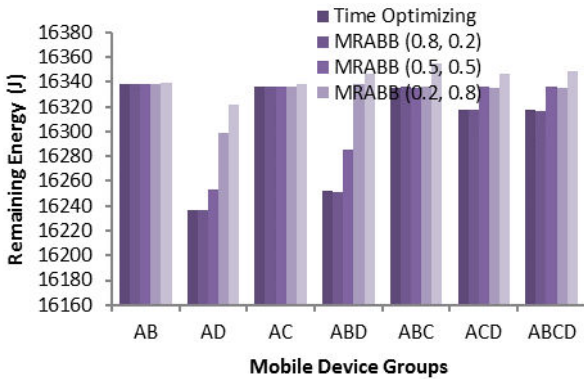
و کم‌ترین طول عمر شبکه را به دست می‌دهد درحالی‌که الگوریتم آگاه از انرژی عکس آن‌ها را به دست می‌آورد. روش پیشنهادی، در این شرایط دستگاه‌های سیار، با در نظر گرفتن هر دو معیار و مدیریت مصالحه میان آن‌ها، وظیفه‌ها را به گونه‌ای توزیع می‌کند که زمان واگذاری و انرژی مصرفی هم‌زمان بهبود داده شوند. الگوریتم MRABB با اوزان  $\{0.8/0.2\}$  و  $\{0.5/0.5\}$  برای دو معیار زمان و انرژی، مقادیری نزدیک به الگوریتم بهینه‌ساز زمان دارد و با اوزان  $\{0.5/0.5\}$  رفتاری بینابین‌تر از خود نشان می‌دهد.



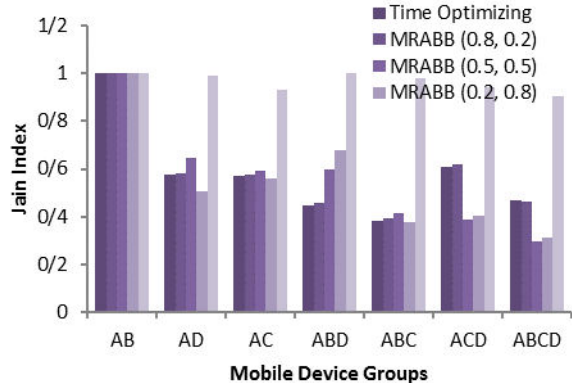
ب) انرژی مصرفی



الف) زمان کل واگذاری



د) کمینه انرژی باقی‌مانده دستگاه‌های سیار



ج) معیار عدالت (Jain)

شکل ۴: تأثیر تعداد و نوع دستگاه‌های سرویس‌دهنده واگذاری بر عملکرد الگوریتم MRABB و مقایسه آن با بهینه‌ساز زمان و آگاه از انرژی

سناریو، علاوه بر ارزیابی کارایی واگذاری و مصرف انرژی دستگاه‌ها، سربار ناشی از اجرای سه الگوریتم تخصیص وظایف نیز مورد بررسی قرار گرفته است. برای این منظور، در شکل ۵-الف و ب، سربار زمان و انرژی به صورت بخشی از نمودار زمان و انرژی کل واگذاری نشان داده شده است.

همان‌طور که در نمودارهای شکل ۵ مشاهده می‌شود، با افزایش تعداد وظیفه‌های برنامه کاربردی در واگذاری، زمان و انرژی مصرفی واگذاری افزایش و کمینه انرژی باقی‌مانده کاهش می‌یابد. همچنین با افزایش تعداد وظیفه‌ها، توزیع وظیفه‌ها متوازن‌تر شده و معیار Jain افزایش یافته است. به علاوه، شیب نمودارهای زمان، انرژی و کمینه انرژی باقی‌مانده، زیاد بوده درحالی‌که شیب نمودار Jain کم است.

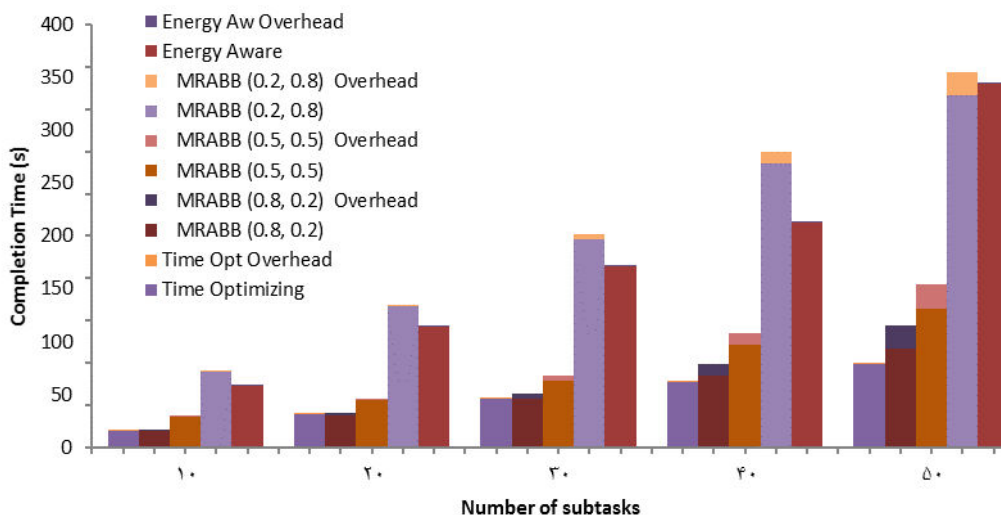
در گروه‌های AC و ABC، الگوریتم آگاه از انرژی علاوه بر اینکه بیش‌ترین زمان واگذاری را به دست می‌دهد، انرژی مصرفی بیش‌تری هم دارد و این آزمایش عدم بهینه بودن این الگوریتم در کمینه‌سازی انرژی در صورت یکسان نبودن وظیفه‌ها را نشان می‌دهد.

۴-۲- تأثیر افزایش تعداد وظیفه‌ها در واگذاری

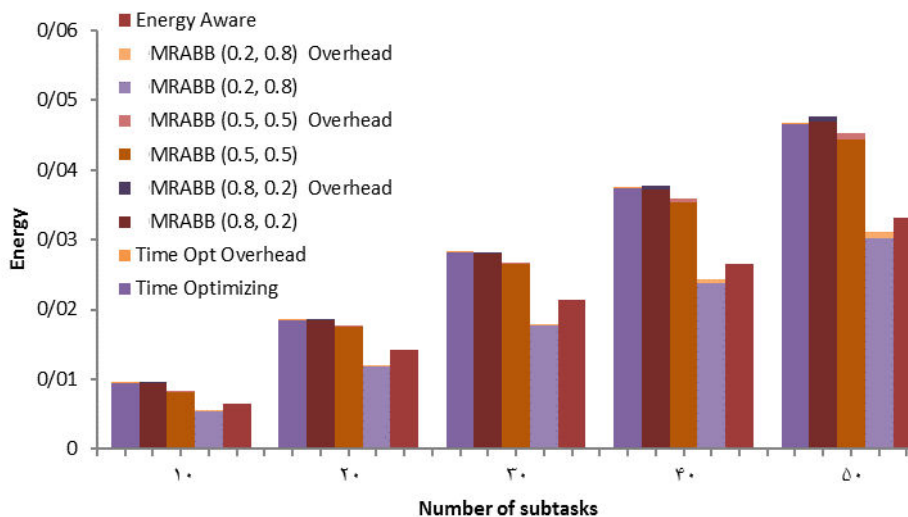
در سناریو دوم، با افزایش تعداد وظیفه‌های یک برنامه کاربردی، تأثیر آن بر عملکرد سه الگوریتم تخصیص وظایف بررسی می‌شود. برای این کار، تعداد عکس‌ها (با حجم ۳۸۰KB) در برنامه تشخیص چهره، بین ۱۰ تا ۵۰ عدد تغییر می‌کند. در این آزمایش، سه دستگاه سیار A، B و D در واگذاری شرکت دارند و دستگاه سیار A متقاضی واگذاری است. در این

کرده است. الگوریتم پیشنهادی با وزن‌های  $\{0.2, 0.8\}$  و  $\{0.8, 0.2\}$  میانگین مصرف انرژی را کمینه کرده و حتی از الگوریتم آگاه از انرژی در بهینه‌سازی مصرف انرژی و افزایش طول عمر شبکه بهتر عمل کرده است. این آزمایش، برتری الگوریتم پیشنهادی در بهینه‌سازی زمان (با وزن‌های  $\{0.2, 0.8\}$  و  $\{0.8, 0.2\}$ )، انرژی (با وزن‌های  $\{0.2, 0.8\}$  و  $\{0.8, 0.2\}$ ) و مصالحه میان آن‌ها (با وزن‌های  $\{0.5, 0.5\}$ ) را نشان می‌دهد.

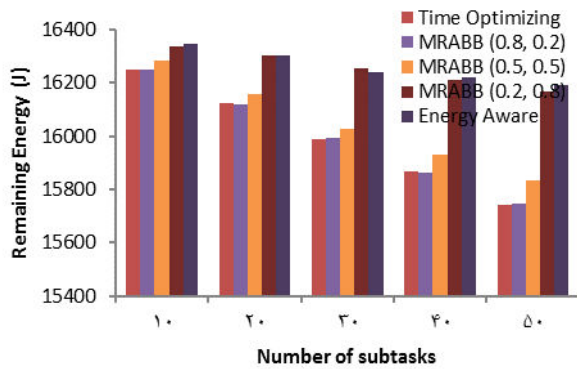
در این حالات، الگوریتم بهینه‌ساز زمان کم‌ترین زمان، بیش‌ترین انرژی و کم‌ترین طول عمر شبکه را به دست می‌دهد در حالی که الگوریتم آگاه از انرژی عکس آن رفتار می‌کند. الگوریتم پیشنهادی MRABB با اوزان  $\{0.2, 0.8\}$  و  $\{0.8, 0.2\}$  برای معیارهای زمان و انرژی، رفتاری بسیار نزدیک به الگوریتم بهینه‌ساز زمان پیدا کرده اما با اوزان  $\{0.5, 0.5\}$ ، الگوریتم پیشنهادی سعی کرده با در نظر گرفتن هر دو معیار، بینابین عمل کند، هر چند مقادیر آن به الگوریتم بهینه‌ساز زمان بیش‌تر گرایش پیدا



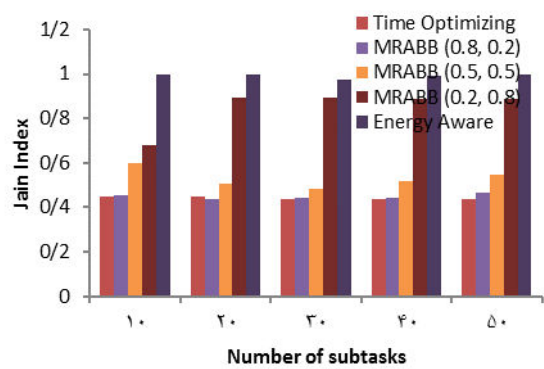
الف) زمان کل واگذاری



ب) انرژی مصرفی



د) کمینه انرژی باقی مانده دستگاه‌های سیار



ج) معیار عدالت (Jain)

شکل ۵: تأثیر افزایش تعداد وظیفه‌های برنامه کاربردی و آگاهی بر عملکرد الگوریتم MRABB و مقایسه آن با بهینه‌ساز زمان و آگاه از انرژی

هریک از این سناریوها دستگاه سیار A درخواست‌کننده‌ی و آگاهی است. مقادیر پارامترهای الگوریتم NSGA-II مورد استفاده در این آزمایش، در جدول زیر نشان داده شده است:

جدول ۵: مقادیر پارامترهای الگوریتم NSGA-II

مقادیر	پارامترها
۵۰	اندازه نسل
۳۰	حداکثر تعداد تولید نسل
۰/۹	احتمال Crossover
$\frac{1}{n}$	احتمال جهش
۲۰	شاخص Crossover
۲۰	شاخص جهش

همان‌طور که در شکل ۶ مشاهده می‌شود، مجموعه نقاط پارتو به‌دست‌آمده از الگوریتم NSGA-II همواره با مجموعه نقاط پارتو بهینه مطابقت پیدا نمی‌کنند. در واقع، الگوریتم‌های تکاملی و ویژگی مکاشفه‌ای و تصادفی دارند. به همین دلیل، فاصله مجموعه نقاط پارتوی کسب شده از مجموعه نقاط پارتوی بهینه اختلاف پیدا می‌کند. همچنین اندازه مجموعه جواب‌های پارتو حاصل از الگوریتم‌های تکاملی کمتر از مجموعه جواب‌های پارتو است، در نتیجه این احتمال وجود دارد که جواب نهایی بهینه از دست برود.

در واقع، نتایج نشان می‌دهد که بین زمان اجرای با کیفیت و تضمین بهینه بودن مجموعه جواب‌های پارتو مصالحه وجود دارد. اگرچه الگوریتم‌های مکاشفه‌ای زمان اجرا را کاهش می‌دهند ولی بهینگی قطعی را تضمین نمی‌دهند. در حالی که روش‌های مبتنی بر شاخه و حد، نزدیک به بهینه بودن نقاط پارتو را تضمین می‌کنند و برای مسئله با اندازه کوچک مانند مسئله حاضر (چون تعداد گره‌های شرکت‌کننده که در همسایگی یک گره در ابر سیار قرار دارند، محدود است) زمان اجرای قابل قبولی دارند. به همین دلیل در این مقاله برای اکتساب جواب‌های بهینه از روش مبتنی بر شاخه و حد در روش MRABB استفاده می‌شود.

در نمودار معیار Jain، الگوریتم آگاه از انرژی بیش‌ترین عدالت را تأمین نموده است. بعد از آن، الگوریتم پیشنهادی با اوزان  $\{0/2\}$  و  $\{0/8\}$  بیش‌ترین میزان عدالت را فراهم می‌آورد و در آخر، الگوریتم بهینه‌ساز زمان کم‌ترین عدالت را تأمین می‌نماید.

نتایج زیر در مورد ارزیابی سربار ناشی از اجرای هر یک از الگوریتم‌های تخصیص وظایف، مشاهده می‌شود: الگوریتم بهینه‌ساز زمان به دلیل ماهیت حریصانه این الگوریتم، کم‌ترین سربار زمان و انرژی مصرفی را برای اجرا دارد. بعد از آن، الگوریتم آگاه از انرژی به دلیل پیچیدگی محاسباتی رابطه انرژی ۲، سربار بیش‌تری دارد ولی در کل این الگوریتم نیز برای اجرا زمان و انرژی کمی مصرف می‌کند. همان‌طور که در شکل ۵ مشاهده می‌شود، الگوریتم پیشنهادی MRABB نسبت به دو الگوریتم دیگر سربار محاسباتی بیش‌تری دارد ولی نسبت به کل و آگاهی این سربار چندان زیاد نیست. به‌طورمثال برای تعداد ۵۰ وظیفه (با وزن‌های  $\{0/5\}$  و  $\{0/5\}$ )، زمان اجرای الگوریتم برابر  $22/18$  ثانیه است که  $14/42$  درصد از زمان کل و آگاهی ( $153/83$  ثانیه) را شامل می‌شود. نکته جالب توجه این است که با وجود پیچیدگی زمانی و مصرف انرژی الگوریتم پیشنهادی، زمان کل و آگاهی و انرژی مصرفی کل در روش پیشنهادی بینابین دو الگوریتم دیگر قرار دارد و مصالحه میان دو معیار زمان و انرژی مصرفی به‌خوبی مدیریت می‌شود.

#### ۳-۴- مصالحه میان زمان و دقت الگوریتم MRABB

در این بخش از آزمایش‌ها، تفاوت دو دسته الگوریتم‌های مبتنی بر شاخه و حد (قطعی) و مبتنی بر مکاشفه‌ای و تکاملی (غیرقطعی) از نظر دقت و تعداد مجموعه جواب‌های پارتوی به‌دست‌آمده، بررسی می‌گردد. همان‌طور که بیان شد، الگوریتم NSGA-II [۲۹] به‌عنوان یکی از الگوریتم‌های تکاملی پرکاربرد برای حل مسائل بهینه‌سازی چندهدفه، با الگوریتم پیشنهادی MRABB مقایسه می‌گردد. مجموعه نقاط پارتو حاصل از الگوریتم MRABB نزدیک به بهینه هستند.

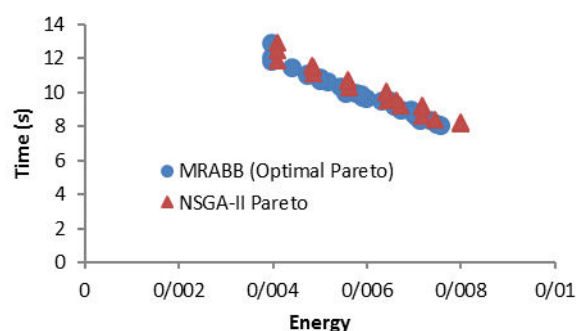
در این سناریو، دستگاه‌های سیار مختلف در و آگاهی شرکت کرده و ۱۰ عکس به‌عنوان ورودی برنامه‌ی کاربردی استفاده شده است. در

ترجیحات از کاربر و استفاده از روش Topsis، بهترین جواب مصالحه‌گر تعیین می‌شود. همچنین برای مدیریت فرایند واگذاری، نمونه‌ای از نرم‌افزار پیشنهادی OMC، پیاده‌سازی و با آزمایش‌های گسترده مورد ارزیابی قرار گرفته است. نتایج آزمایش‌ها نشان می‌دهند روش تخصیص وظایف پیشنهادی در مقایسه با روش بهینه‌ساز زمان و آگاه از انرژی، با در نظر گرفتن هر دو معیار زمان واگذاری و انرژی مصرفی و مدیریت مصالحه‌ی میان آن‌ها، سعی می‌کند وظیفه‌ها را به‌گونه‌ای توزیع کند که این دو معیار همزمان کمینه شوند.

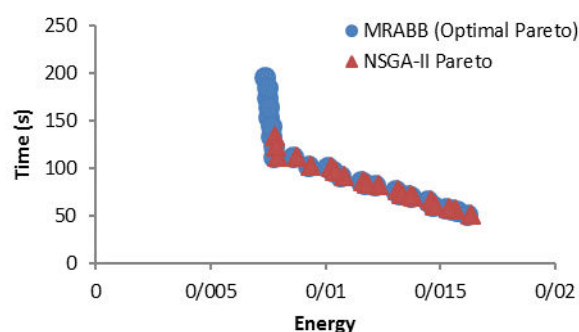
با توجه به تحرک بالای دستگاه‌های سیار، احتمال خارج شدن دستگاه‌ها از محدوده‌ی گره‌ی مبدأ در حین واگذاری، بالا می‌باشد. بررسی تحرک گره‌ها و تصمیم‌گیری در مورد تخصیص وظایف با توجه به در دسترس بودن گره‌ها، یکی از کارهای تحقیقاتی است که می‌تواند در ادامه این پژوهش انجام گیرد. همچنین، طراحی یک مکانیزم مدیریت خرابی و مشکلات احتمالی در حین واگذاری می‌تواند به‌عنوان یک کار پژوهشی مورد توجه قرار گیرد.

### مراجع

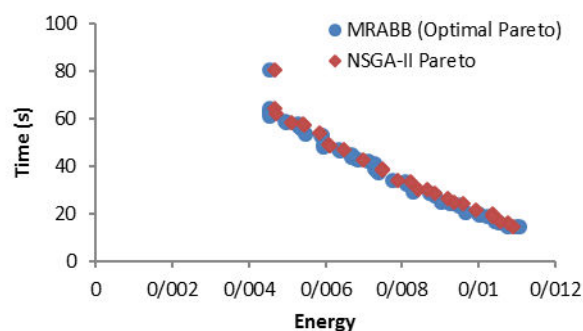
- [1] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84-106, 2013.
- [2] G. Huerta-Canepa, and D. Lee, "A virtual cloud computing provider for mobile devices," *1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond, MCS'10*, pp. 6-11, ACM, 2010.
- [3] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *Pervasive Computing, IEEE*, vol. 8, no. 4, pp. 14-23, 2009.
- [4] G. F. Huerta Canepe, *A Context-aware Application Offloading Scheme for a Mobile Peer-to-peer Environment*, Ph.D. dissertation, Department of Information and Communication Engineering, KAIST, South Korea, 2012.
- [5] E. Cuervo, et al., "MAUI: making smartphones last longer with code offload," *8th International Conference on Mobile Systems, Applications, and Services, MobiSys'10*, pp. 49-62, 2010.
- [6] M. D. Kristensen, "Scavenger: Transparent development of efficient cyber foraging applications," *IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 217-226, 2010.
- [7] R. C. Marin, and C. Dobre, "Reaching for the clouds: contextually enhancing smartphones for energy efficiency," *2nd ACM Workshop on High Performance Mobile Opportunistic Systems*, pp. 31-38, 2013.
- [8] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura, "Serendipity: enabling remote computing among intermittently connected mobile devices," *13th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 145-154, 2012.
- [9] A. Mtibaa, A. Fahim, K. A. Harras, and M. H. Ammar, "Towards resource sharing in mobile device clouds: Power balancing across mobile devices," *2nd ACM SIGCOMM Workshop on Mobile Cloud Computing*, pp. 51-56, 2013.
- [10] J. H. Chang, and L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks," *19th Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM*, vol. 1, pp. 22-31, 2000.



الف) گره‌های ابر سیار ABCD،  $k=4$



ب) گره‌های ابر سیار ABD،  $k=3$



ج) گره‌های ابر سیار ABD،  $k=3$

شکل ۶: مقایسه مجموعه نقاط پارتو بهینه حاصل از الگوریتم MRABB با مجموعه پارتو به‌دست‌آمده از الگوریتم NSGA-II

### ۵- نتیجه‌گیری و کارهای آینده

تخصیص وظایف بین دستگاه‌های سیار مجاور که نقش فراهم‌کنندگان منابع را بر عهده دارند، یکی از مسائل مهم و چالش‌برانگیز در رایانش ابری سیار می‌باشد. هدف تخصیص وظایف در این تحقیق، کمینه‌سازی همزمان دو معیار زمان و انرژی مصرفی دستگاه‌های سیار است به‌طوری که برخی قیود کیفیت سرویس برآورده شوند. در این مقاله، مسئله‌ی تخصیص وظایف در ابر سیار به صورت یک بهینه‌سازی چندهدفه قیددار مدل و یک روش دو مرحله‌ای آن پیشنهاد شد. در گام اول، الگوریتمی مبتنی بر شاخه و حد چندهدفه (MRABB) برای یافتن مجموعه جواب‌های پارتو طراحی شده است. در گام دوم، با گرفتن بردار

- IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182-197, 2002.
- [۳۰] محمدامیر عباسیان و حسین نظام‌آبادی‌پور، «الگوریتم جستجوی گرانشی چندهدفه مبتنی بر مرتب‌سازی جبهه‌های مغلوب‌نشده»، *مجله مهندسی برق دانشگاه تبریز*، دوره ۴۱، شماره ۱، صفحات ۶۸-۸۰، ۱۳۹۰.
- [31] E. L. Lawler, and D. E. Wood, "Branch-and-bound methods: A survey," *Operations Research*, vol. 14, no. 4, pp. 699-719, 1966.
- [11] S. Ghasemi-Falavarjani, M. Nematbaksh, and B. Shahgholi Ghahfarokhi, "A multi-criteria resource allocation mechanism for mobile clouds," *International Symposium on Computer Networks and Distributed Systems (CNSD 2013)*, Communications in Computer and Information Science (CCIS), Springer International Publishing, vol. 428, pp. 145-154, Tehran, Iran, 2014.
- [۱۲] محمدجواد اصغرپور، *تصمیم‌گیری چندمعیاره*، چاپ دوم، انتشارات دانشگاه تهران، ۱۳۸۱.
- [13] G. Wei, A. V. Vasilakos, Y. Zheng, and N. Xiong, "A game-theoretic method of fair resource allocation for cloud computing services," *The Journal of Supercomputing*, vol. 54, no. 2, pp. 252-269, 2010.
- [14] F. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, no. 1, pp. 33-37, 1997.
- [15] M. R. Rahimi, N. Venkatasubramanian, S. Mehrotra, and A. V. Vasilakos, "On optimal and fair service allocation in mobile cloud computing," *arXiv preprint arXiv:1308.4391*, [Online] Available online at: <http://arxiv.org/abs/1308.4391>, 2013.
- [16] M. Ehrgott, *Multicriteria Optimization*, 2<sup>nd</sup> ed., Springer, Berlin, 2005.
- [17] M. Gavanelli, "An implementation of Pareto optimality in CLP (FD)," *CPAI-OR-International Workshop on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimisation Problems*, pp. 49-64, 2002.
- [18] V. Kariwala, and Y. Cao, "Branch and bound method for multiobjective pairing selection," *Automatica*, vol. 46, no. 5, pp. 932-936, 2010.
- [19] G. H. Tzeng, and J. J. Huang, *Multiple Attribute Decision Making: Methods and Applications*, CRC Press, 2011.
- [20] C. L. Hwang, and K. Yoon, *Multiple Attribute Decision Making Methods and Applications: A State-of-art Survey*, New York, Springer-Verlag, 1981.
- [21] M. D. Kristensen, *Scavenger-mobile Remote Execution*, DAIMI Report Series, vol. 37, no. 587, 2008.
- [22] B. G. Chun, *et al.*, "Clonecloud: elastic execution between mobile device and cloud," *6th ACM Conference on Computer Systems*, pp. 301-314, 2011.
- [23] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, "AIOLOS: Middleware for improving mobile application performance through cyber foraging," *Journal of Systems and Software*, vol. 85, no. 11, pp. 2629-2639, 2012.
- [24] *NBench*, [Online] Available online at: <http://www.tux.org/~mayer/linux/bmark.html/>.
- [25] L. Zhang, *et al.*, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," *8th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis*, pp. 105-114, 2010.
- [26] *Joulmeter*, [Online], Available online at: <http://research.microsoft.com/>.
- [27] R. Jain, D. M. Chiu, and W. R. Hawe, *A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System*, Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA, 1984.
- [28] C. M. Fonseca, and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evolutionary computation*, vol. 3, no. 1, pp. 1-16, 1995.
- [29] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II,"

زیرنویس‌ها

- ۱ Offload
- ۲ Offloading in Mobile Cloud (OMC)
- ۳ Scavenger
- ۴ Cyber foraging
- ۵ Serendipity
- ۶ Trade off
- ۷ Offloading Mobile Cloud Framework (OMCF)
- ۸ Compromise
- ۹ Multi-objective Resource Allocation Branch and Bound algorithm
- ۱۰ Technique for Order Preference by Similarity to Ideal Solution
- ۱۱ Offloading in Mobile Cloud
- ۱۲ Context manager
- ۱۳ Non-dominated Sorting
- ۱۴ Crowding-distance