

# استفاده از الگوریتم خفاش به منظور کسب سود حداکثری از تخصیص و قیمت گذاری منابع زیرساخت ابری در بازار حراج دوطرفه

فاطمه اکبری<sup>۱</sup>، محمد رضا میبدی<sup>۲</sup>

<sup>۱</sup> دانشجوی کارشناسی ارشد، دانشگاه آزاد اسلامی واحد قزوین

<sup>۲</sup> عضو هیأت علمی دانشگاه صنعتی امیر کبیر

[fakbari.co@gmail.com](mailto:fakbari.co@gmail.com)<sup>۱</sup>

[mmeybodi@iau.ac.ir](mailto:mmeybodi@iau.ac.ir)<sup>۲</sup>

چکیده- محاسبات ابری یک رویکرد جدید جهت ارائه منابع محاسبانی از سطح زیرساخت تا نرم افزار به عنوان خدمت است که هنوز با چالش‌های بسیاری روبروست. یکی از چالش‌های مهمی که ارائه دهندگان سرویس‌های ابری با آن مواجه هستند، تخصیص مؤثر منابع زیرساخت ابری با هدف کسب سود بیشتر است. طرح‌های قیمت گذاری ایستا پاسخگوی نیازهای مبتنی بر عرضه و تقاضای بازار ابر نبوده و پژوهشگران به دنبال یک روش منعطف و پویا با سربار کاری کم هستند که بتوانند چنین شرایطی را مدیریت نمایند. این مقاله یک روش فراابتکاری با الگوریتم خفاش اصلاح شده به منظور حل مسئله تخصیص و قیمت گذاری منابع ابری در بازار حراج دوطرفه با هدف کسب سود بیشتر از ابر ارائه می‌دهد که در فرمول سرعت با حالت استاندارد الگوریتم خفاش تفاوت دارد. نتایج شبیه سازی مدل پیشنهادی در شبیه ساز قدرتمند کلودسیم با نتایج الگوریتم خفاش استاندارد، ژنتیک، مورچگان و رقابت استعماری مورد مقایسه قرار گرفت و نشان‌دهنده این است که الگوریتم خفاش پیشنهادی نتایج بسیار خوبی را در توابع معیار سود حاصل از ابر، زمان پاسخگویی و مصرف انرژی ارائه نموده است. با توجه به  $NP-Hard$  بودن مسئله و پیچیدگی‌های فراوان پیاده سازی ساختار مدل حراج دوطرفه در دنیای واقعی، مسائل باز زیادی پیش روی پژوهشگران وجود دارد.

کلید واژه- محاسبات ابری، بازار حراج دوطرفه، الگوریتم خفاش، تابع معیار سود.

## ۱- معرفی

ها اهمیت ندارد به اتمام برسد و از طرف دیگر اگر قیمت را بالا در نظر بگیرند ممکن است کاربران استفاده از این منابع گران را کنار بگذارند و در نتیجه این امر باعث اتلاف منابع و کاهش کارایی می‌گردد. مدیریت درست این سرویس‌ها به طوری که بتوان با سود بیشتری به ارائه سرویس با کیفیت مورد نظر پرداخت، امری ضروری می‌باشد. از این رو مطرح شدن مدل قیمت گذاری پویا سبب محبوبیت بیشتری نزد کاربران و سرویس دهندگان شده است. یکی از مدل‌های محبوب بازار در تجارت حراج است که با ایجاد بازار عرضه و تقاضا و نیز امکان تخصیص منابع بطور همزمان، با ایجاد یک محیط رقابتی موجب افزایش سود فروشندگان و کاهش زمان پاسخگویی به کاربران می‌شود. به این ترتیب می‌تواند یکی از مناسب‌ترین مدل‌ها برای قیمت گذاری منابع ابری باشد. اولین بار آمازون EC2 یک مکانیزم حراج مبتنی بر بازار را برای تأمین ماشین مجازی مشتریان ارائه نمود، اما قیمت‌ها بازار محور نبود. با حرکت به

بازار خدمات ابری به سرعت در حال رشد می‌باشد. در زیرساخت به عنوان سرویس که یکی از مدل‌های ارائه شده است، ارائه دهندگان نمونه‌های ماشین مجازی از پیش پیکربندی با تعداد در حال افزایش را در طول سال عرضه می‌کنند. امروزه بزرگترین سرویس دهندگان ابری مانند آمازون و گوگل، منابع محاسباتی خود را با قیمت ایستا ارائه می‌دهند که با پیچیده شدن سرویس‌ها و سیل درخواست‌هایی که باعث نوسان زیاد عرضه و تقاضا شده‌اند دیگر کارایی لازم را نداشته و پژوهشگران به دنبال یک روش منعطف و پویا با سربار کاری کم هستند که بتوانند چنین شرایطی را مدیریت نمایند. مشکل اصلی این است که ارائه دهندگان چگونه قیمت را تعیین کنند؟ اگر قیمت را پایین در نظر بگیرند ممکن است منابع توسط کاربرانی که قیمت برای آن-

سمت قیمت گذاری مبتنی بر بازار، حراج برای تخصیص منابع ابری اخیراً طراحی شده است [1].

بطور معمول، زیرساخت به عنوان سرویس در فرمی از ماشین-های مجازی ارائه شده است که توسط تعدادی از ویژگی‌های سطح پایین مانند پردازنده، حافظه، ذخیره سازی، شبکه و ... مشخص می‌شود. با توجه به ماهیت ناهمگون خدمات ابر زیرساخت و دیگر نیازمندی‌های بازار ابر، طراحی مکانیزم بازار محاسبات ابری کار دشواری است [2].

استفاده از روش‌های فراابتکاری به هدف حل مسائل بهینه سازی در چند دهه اخیر بسیار مورد توجه بوده است. بسیاری از این روش‌ها از رویه‌های جستجوی محلی بهره می‌برند و برای اینکه در نقاط بهینه محلی باقی نمانند از ترفندهای ویژه‌ای استفاده می‌کنند [3]. با توجه به NP-Hard بودن مسئله تخصیص و قیمت گذاری منابع ابری که طبق تعریف، روشی برای دستیابی به جواب نهایی آن‌ها در زمان معقول وجود ندارد، ما برای حل مسئله از یک الگوریتم خفایش پیشنهادی استفاده نمودیم.

تحقیقات انجام شده با فرض حراج منابع ناهمگون زیرساخت ابری و سه تابع معیار سود ابر، زمان پاسخگویی و مصرف انرژی صورت گرفته است. ما نتایج بدست آمده از مدل پیشنهادی را با الگوریتم خفایش استاندارد، ژنتیک، مورچگان و رقابت استعماری مقایسه کردیم. طبق آزمایشات می‌توان اظهار داشت که مدل پیشنهادی ما در هر سه تابع معیار نتایج قابل قبولی بدست آورده است.

در ادامه این مقاله در بخش ۲ تاریخچه مهم‌ترین کارهای مرتبط و در بخش ۳ جزئیات مدل پیشنهادی ارائه شده است. در بخش ۴ شبیه سازی و ارزیابی و مقایسه مدل با سایر الگوریتم‌ها ارائه می‌گردد. نتیجه گیری نهایی در بخش ۵ مطرح می‌شود.

## ۲- کارهای انجام شده

برخی کارها طرح‌های قیمت گذاری ثابت و پویا را مقایسه نمودند [4]، [5]. Hong و همکاران از مدل مبتنی بر مسدود سازی منابع به عنوان یک استراتژی مناسب و مالی جذاب جهت بهبود درآمد در هزینه مشتریان ابر استفاده نمودند [6]. Mihailescu و Teo یک طرح قیمت گذاری پویا برای ابر فدرال را جایی که منابع روی تعدادی از ارائه‌دهندگان خدمات ابری به اشتراک گذاشته شده‌اند، تعریف کردند [7]. Qian Wang و همکاران مدل مبتنی بر حراج را برای منابع محاسباتی ابری پیشنهاد دادند [8]. Hao Li و همکاران مدل عامل بانک ابر را به عنوان یک آژانس منابع از

دیدگاه جهانی، تجزیه و تحلیل نموده و برای همه شرکت کنندگان یک الگوریتم بروز رسانی تکرارشونده قیمت فراهم می‌کند [9]. برخی از مقالات بر روی قیمت گذاری نرم افزار به عنوان خدمات ابری تمرکز نموده‌اند [10] و [11]. Mario و Guitart یک مدل ژنتیکی برای قیمت گذاری در بازارهای ابر رایانه ارائه دادند [12]. Wei wang و همکاران حراج پویایی را که به سرعت با تغییرات بازار وفق می‌یابد در حالی که ارائه خدمات تضمین می‌شود ارائه نمودند [13]. Adel nadjaran و همکاران حراج حسادت آزادی را ارائه نمودند که با احتمال بالا راستگوست و سود نزدیک به بهینه را برای ارائه دهنده ابر تولید می‌کند. حراج تحت شرایط مشابه به بازار نقطه EC2 عمل می‌کند [14]. Xiaoxi Zhang و همکاران یک مکانیزم حراج ماشین مجازی آنلاین را طراحی کردند که هزینه‌های سرور در محاسبه سود ارائه دهنده ابر از پرداخت‌های نهایی کاربران کسر گردیده است [1]. در [15] نویسندگان رویکرد حریصانه Lehmann را به عنوان کران پایین بهینگی در مکانیزم جستجو تپه نوردی به منظور بهبود کارایی تخصیص منبع از مکانیزم حراج مورد استفاده قرار دادند. در [16] و [17] Sergie Chichin و همکاران سناریوی ابر واقعی را در نظر گرفته و مکانیزم‌های حریصانه برای حراج ترکیبی با قیمت‌های رزرو فروشنده را توسط الگوریتم کوله پشتی در بازارهای ابر یک طرفه با در نظر گرفتن اصل صداقت در مزایده تعریف کردند. در [2] و [18] آن‌ها روی طراحی مکانیزم-های تخصیص دو طرفه مؤثر مناسب برای بازار باز تمرکز نمودند و از آنجا که مطالعات نظری بدون پیاده سازی در دنیای واقعی در بازارهای باز ابر وجود دارد، آن‌ها عملکرد تخصیص خود را در مجموعه‌ای از آزمایشات بررسی نمودند.

## ۳- مدل پیشنهادی

### ۳-۱ ملزومات

ما یک مدل حراج برای مبادله خدمات ابری زیرساخت که شامل سه ساختار (بازار، کالاهای مبادله شده و شرکت کنندگان در بازار) است را در نظر می‌گیریم.

بازار: یک واسط برای تجارت و هدایت روند ارائه خدمات ابر زیرساخت است. در این کار ما مسئله تخصیص و قیمت‌گذاری را مبتنی بر بازار فرض می‌کنیم چرا که تصمیم‌گیری در مدل‌های تجاری مبتنی بر بازار هماهنگ با پویایی محیط است.

کالا: خدمات زیرساخت ابری هستند که با توجه به عرضه و تقاضا در میان شرکت کنندگان حراج توزیع می‌شوند. ارائه دهنده

خدمات ابر زیرساخت منابع را در قالب ماشین‌های مجازی ارائه می‌دهد. چنانچه همه کالاها از یک نوع باشد به آن کالای همگن و اگر چند نوع کالا مورد مبادله قرار گیرد به آن ناهمگن گویند. در این کار ما سه نوع کالا ( $k=3$ ) شامل پردازنده، حافظه و هارد دیسک را برای تجارت در نظر می‌گیریم.

شرکت کنندگان: به منظور دسترسی یا ارائه خدمات جهت شرکت در فعالیت تجاری به بازار متصل می‌شوند. شرکت کنندگان بازار هم می‌توانند مصرف کننده خدمات باشند و هم ارائه دهنده که یا خدمات را می‌خرند و یا می‌فروشند. چنانچه رقابت از یک طرف (خریدار یا فروشنده) باشد به آن یک طرفه و در صورتی که رقابت از هر دو طرف خریدار و فروشنده باشد به آن دو طرفه گویند که ما در این کار فرض می‌کنیم بازار حراج از نوع دوطرفه باشد.

## ۳-۲ مدل خفاش پیشنهادی

الگوریتم خفاش یکی از الگوریتم‌های فراابتکاری الهام گرفته از طبیعت است که در سال ۲۰۱۰ توسط آقای یانگ معرفی گردید [19]. این الگوریتم بر مبنای استفاده از خاصیت انعکاس صدا توسط خفاش‌هاست. خفاش‌ها مسیر و محل دقیق طعمه خود را بوسیله فرستادن امواج صوتی و دریافت بازتاب آن، پیدا می‌کنند. زمانی که امواج صوتی به سمت فرستنده امواج (خفاش) باز می‌گردد، این پرنده می‌تواند یک تصویر صوتی از موانع روبرو و محیط اطراف خود رسم کند و محیط اطراف را به خوبی ببیند. منطق الگوریتم بدین صورت است که هر خفاش مجازی با سرعت برابر  $V_i$  بطور تصادفی پرواز می‌کند. مکان آن یا  $X_i$  جواب نهایی این الگوریتم است. یک خفاش در حین جستجویش برای یافتن شکار طول موج صدا ( $A_i$ ) و نرخ انتشار پالس ( $r_i$ ) را تغییر می‌دهد. هم‌چنین جستجو توسط گام تصادفی تقویت می‌شود.

هر خفاش به عنوان یک جواب در نظر گرفته می‌شود و دارای دو المان سرعت و مختصات است. یک تابع هزینه در این روش تعریف می‌گردد که با کمک آن کاندیدها ارزیابی می‌شوند و سرعت و مختصات آن‌ها به روز رسانی می‌گردد. در ابتدا سرعت و وضعیت هر ذره به صورت تصادفی انتخاب می‌گردد. هر خفاش از سرعت و مختصات برای ارزیابی جواب‌ها و هم‌چنین میزان واگرایی جواب‌ها از یکدیگر استفاده می‌کند. در الگوریتم خفاش استاندارد سرعت خفاش‌ها از فرمول زیر محاسبه می‌شود:

$$v_i^t = v_i^{t-1} + (x_i^t - x^*)f_i \quad ۱$$

$x^*$  مختصات خفاش بهینه سراسری در زمان  $t$ .

$x_i^t$  مختصات خفاش  $i$  ام در زمان  $t$ .

$v_i^t$  سرعت خفاش  $i$  ام در زمان  $t$ .

فرکانس صدا یا  $f_i$  به صورت زیر تعریف شده است:

$$f_i = f_{min} + (f_{max} - f_{min})\beta \quad ۲$$

$f_{max}$  و  $f_{min}$  مقادیر کمینه و بیشینه فرکانس هستند.

$\beta$  نیز مقداری بین [0,1] است.

$f_i$  مقدار فرکانس خفاش  $i$  است.

مختصات هریک از خفاش‌ها نیز با فرمول‌های زیر به روز رسانی می‌گردد [20]:

$$x_i^t = x_i^{t-1} + v_i^t \quad ۳$$

$$x_{new} = x_{old} + \epsilon A^t \quad ۴$$

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)] \quad ۵$$

$$A_i^{t+1} = \alpha A_i^t \quad ۶$$

الگوریتم خفاش مزایای زیادی دارد. یکی از مزایای کلیدی آن توانایی همگرایی سریع در مرحله ابتدایی بوسیله سوئیچ از اکتشاف به استخراج است. این ویژگی باعث می‌شود این الگوریتم، یک الگوریتم تأثیرگذار برای کاربردهایی مثل دسته بندی و در کل زمانی که یک راه حل سریع نیازمند است، باشد. هرچند که با اجازه دادن به الگوریتم به سوئیچ سریع به مرحله استخراج، با تغییر سریع  $A$  و  $r$ ، بعد از چند مرحله ابتدایی الگوریتم دچار رکود می‌شود. به منظور بهبود کارایی الگوریتم، متدها و استراتژی‌های زیادی مطرح شده‌اند تا تنوع راه حل را افزایش دهند که منجر به تولید نسخه‌های مختلفی از الگوریتم خفاش شده است. ما به منظور بهبود الگوریتم خفاش استاندارد فاکتور وزن را به صورت زیر در نظر گرفته و سپس آن را در فرمول سرعت ضرب نمودیم.

$$W = \frac{(max\ iter - iter)}{max\ iter} \times (W_{max} - W_{min}) + W_{min} \quad ۷$$

$$v_i^t = W [v_i^{t-1} + (x_i^t - x^*)f_i] \quad ۸$$

فاکتور وزن مطابق فرمول ۷ تعریف شده است و فرمول ۸ جایگزین فرمول ۱ الگوریتم خفاش استاندارد می‌شود. شبه کد الگوریتم خفاش وزن دار پیشنهادی به شرح ذیل می‌باشد:

### WBA Algorithm

Objective function  $f(x)$ ,  $x = (x_1, x_2, \dots, x_d)^T$

Initialize the bat population  $x_i$  ( $i=1,2,\dots,n$ ) and  $V_i$

Define pulse frequency  $f_i$  at  $x_i$

Initialize pulse rates  $r_i$  and the loudness  $A_i$

While ( $t < \text{Max number of iteration}$ )

Generate new solutions by adjusting frequency,

Calculate  $W$  [equation (7)]

۳-پیشنهادهای با توجه به قیمت پیشنهادی به صورت نزولی مرتب می‌شوند.

۴-منابع ارائه شده با توجه به قیمت به صورت صعودی مرتب می‌شوند.

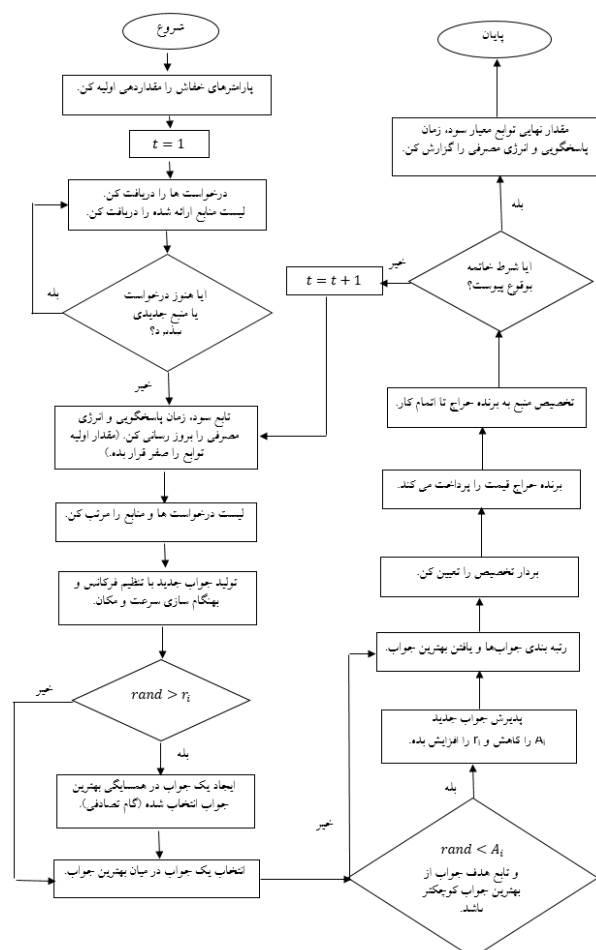
۵-حراج کننده عملیات حراج را انجام می‌دهد و برنده را با توجه به پارامترهای الگوریتم خفاش تعیین می‌کند.

۶-برنده قیمت را پرداخته و منبع به درخواست برنده حراج تخصیص می‌یابد و تا اتمام کار در اختیار آن است.

۷-پس از انجام کار، منبع آزاد شده و نتایج به پیشنهاد دهنده ارسال می‌گردد.

۸-عملیات تا شرط توقف ادامه می‌یابد. (حداکثر مراحل حراج و یا اینکه درخواست جدید قابل انجامی وجود نداشته باشد).

شکل ۱ فلوچارت الگوریتم پیشنهادی را نشان می‌دهد.



شکل ۱: فلوچارت الگوریتم پیشنهادی.

and updating velocities and locations/solutions [equations (2 – 3 and 8)]

If ( $rand > r_i$ )

Select a solution among the best solutions

Generate a local solution around the selected best solution

end if

Generate a new solution by flying randomly

If ( $rand < A_i \& f(x_i) < f(x^*)$ )

Accept the new solutions

Increase  $r_i$  and reduce  $A_i$

end if

Rank the bats and find the current best  $X^*$

end while

Postprocess results and visualization

### ۳-۳ مدل کردن مسئله

در روش پیشنهادی، درخواست‌های کاربران به عنوان خفاش در نظر گرفته شده است. پردازش الگوریتم خفاش ابتدا با یک جمعیت از درخواست‌ها آغاز می‌شود. هر خفاش از گذشته خود و دیگر اعضا بهره می‌برد. خفاش یک ماتریس است که اندیس‌ها شماره کارها و مقدار درایه‌های آن، مقدار ماشین‌ها را نشان می‌دهد. هدف اصلی این پژوهش، انجام همه کارهای ممکن با کسب سود بیشتر ابر است که هم رضایت کاربران و هم رضایت ارائه دهندگان منابع ابری را شامل شود. وظیفه الگوریتم شناسایی بهترین بردار تخصیص کاربر به ارائه دهنده خدمات ابری جهت انجام کارهاست.

در گام اول هر مشتری و ارائه دهنده پیشنهاد خود را به حراج کننده می‌فرستند. بعد از تعیین برنده‌ها، در گام دوم حراج کننده منابع ارائه دهنده را به مشتریان برنده حراج تخصیص می‌دهد. در گام سوم عملیات قیمت گذاری انجام شده و مشتری قیمت تعیین شده را به ارائه دهنده می‌پردازد. حراج کننده وظیفه فروش و کنترل بر روی درخواست‌ها را برعهده دارند. در مدل حراج در محاسبات ابری، قبل از شروع حراج، پارامترهایی نظیر تعداد منابع، تعداد کاربران، تعداد فروشندگان، حداکثر تعداد مراحل حراج مشخص می‌باشد. این عوامل در طول اجرا تغییر نمی‌کنند و به صورت ایستا می‌باشند. مراحل اجرای کار به شرح ذیل است:

۱-ارائه دهندگان منابع موجود خود ( $k=1, \dots, n$ ) را با قیمت

پیشنهادی خود اعلام می‌کنند.

۲-پیشنهاد دهندگان درخواست‌های خود را ( $j=1, \dots, N$ ) و

قیمتی که حاضر به پرداخت برای انجام کار هستند،

ارسال می‌کنند.

#### ۴-۳ توابع معیار

ما کارمان را با سه تابع هدف محاسبه سود ابر و زمان پاسخگویی و انرژی صرف شده بررسی کردیم.

##### - تابع سود

سود حاصل از ابر برابر با قیمتی است که کاربران به ازای کاری که درخواست می کنند، پرداخت می نمایند منهای هزینه ای که ارائه دهنده بابت انجام کار متقبل می شود. تابع هزینه درخواست  $j$  با فرمول ۹ محاسبه می شود.  $C_i$  هزینه هر ماشین مجازی از نوع  $i$  است.  $k$  تعداد انواع ماشین مجازی و  $n_i$  تعداد ماشین مجازی از نوع  $i$  می باشد.  $x_{j,i,r}$  درایه ماتریس دودویی تخصیص است که اگر  $r$ امین ماشین مجازی از نوع  $i$  به درخواست  $j$  تخصیص یافته باشد، مقدارش یک و در غیر این صورت صفر است. درآمد حاصل از هر درخواست  $j$  با فرمول ۱۰ محاسبه می شود که  $Pricebidder$  مبلغی است که کاربر به ازای کاری که درخواست می کند، می پردازد. از آنجایی که هدف مسئله بیشینه نمودن سود کل ابر است، برای محاسبه سود کل از فرمول ۱۱ استفاده نمودیم:

$$Cost_j = \sum_{i=1}^k \sum_{r=1}^{n_i} (x_{j,i,r} \times C_i) \quad 9$$

$$Revenue_j = Pricebidder - Cost_j \quad 10$$

$$Maximize: \quad 11$$

$$Profit = \sum_{j=1}^N Revenue_j$$

$$s.t. \quad (1) \quad \forall i: \sum_{j=1}^N \sum_{r=1}^{n_i} x_{j,i,r} \leq n_i \quad 12$$

$$s.t. \quad (2) \quad \forall j, \forall i: x_{j,i,r} \in \{0, 1\} \quad 13$$

محدودیت اول اجازه نمی دهد که تعداد ماشین های مجازی اختصاص یافته به درخواست های کاربران، از تعداد مجموع آنها که برای هر نوع  $i$  برابر با  $n_i$  است، تجاوز کند. محدودیت دوم بیان می کند که اگر  $r$ امین ماشین مجازی از نوع  $i$  به درخواست  $j$  تخصیص یافته باشد، مقدارش یک و در غیر این صورت صفر است.

##### - تابع زمان پاسخگویی

تابع هدف دوم زمان پاسخگویی به درخواست هاست. در محاسبه زمان محاسباتی هر درخواست کاربر  $time_j$ ، علاوه بر زمانی که جهت اجرای کار صرف شده  $time_{execution}$ ، مدت زمانی را که کار در صف انتظار بوده  $time_{hold}$ ، نیز لحاظ گردیده است. برای

محاسبه زمان پاسخگویی برای درخواست  $j$  در ابر از فرمول ۱۴ استفاده نمودیم. برای محاسبه زمان کل پاسخگویی به تمامی درخواست های ممکن از فرمول ۱۵ استفاده می کنیم:

$$time_j = time_{execution} + time_{hold} \quad 14$$

$$time_{all} = \sum_{j=1}^N time_j \quad 15$$

##### - تابع انرژی

مقدار انرژی مصرفی به وسیله گره های محاسباتی در مراکز داده غالباً به وسیله پردازنده، حافظه، دیسک ذخیره سازی، منبع تغذیه و سیستم های خنک کننده تعیین می شود. مطالعاتی که اخیراً انجام شده است نشان داده که انرژی مصرفی به وسیله سرورها به طور تقریبی رابطه خطی با بهره وری پردازنده دارد.

شبیه ساز کلاسیک برای محاسبه توان از مدل ریاضی DVFS بهره می برد. روش DVFS یک روش بهینه برای کنترل مصرف توان پردازنده ها محسوب می گردد. این روش را نمی توان بر روی مؤلفه های دیگر سیستم هم چون حافظه یا شبکه اعمال کرد. تجانس ولتاژ پویا و فرکانس (DVFS) فرصت بزرگی را ارائه می دهد تا به طور چشمگیری کاهش انرژی/ مصرف برق را با تنظیم ولتاژ و فرکانس یک سیستم با توجه به ویژگی تغییر در وظایف سنگین خود ارائه دهد. از آنجا که پردازنده ها مهم ترین فاکتور مصرف انرژی در مراکز داده ای محسوب می شوند، کاهش توان مصرفی پردازنده ها با استفاده از روش DVFS یک نقش حیاتی در میان روش های ارائه شده برای کاهش مصرف توان مراکز داده ای است.

مصرف توان پردازنده شامل مصرف توان دینامیک و استاتیک است. بر این اساس، کل توان مصرفی یک پردازنده با فرمول ۱۶ مدل می شود.  $\beta_i$  و  $\alpha_i$  به ترتیب بیان کننده مصرف توان بی کاری و ضریب تناسبی هستند.  $f(t)$  نشان دهنده سطح فرکانسی پردازنده در لحظه  $t$  است. هر پردازنده مجهز به روش DVFS است که می تواند سطح فرکانسی پردازنده را در یکی از نقاط عملیاتی بین فرکانس بیشینه و فرکانس کمینه، تنظیم نماید. مصرف انرژی پردازنده با فرمول ۱۷ محاسبه می شود:

$$P_i = \beta_i + \alpha_i f(t)^3 \quad 16$$

$$E_i = \int_{t_0}^{t_1} P_i(f(t)) dt \quad 17$$

$$Energy Consumption = \int_{t_0}^{t_1} P_i(f(t)) dt \quad 18$$

جدول ۱: نشانه‌ها و تعریف‌ها

تعریف	نشانه
تعداد انواع منبع ارائه شده فروشنده‌ها	$k$
تعداد درخواست‌های کاربران	$N$
تعداد ماشین مجازی از نوع $i$	$n_i$
هزینه هر ماشین مجازی از نوع $i$	$C_i$
هزینه انجام کار $j$	$Cost_j$
درآمد حاصل از انجام کار $j$	$Revenue_j$
قیمتی که کاربر برای دریافت منابع درخواستی خود پیشنهاد می‌دهد.	$Price_{bidder}$
سود کل ابر	$Profit$
مدت زمانی که صرف اجرای کار $j$ می‌شود.	$time_{execution}$
مدت زمانی را که کار $j$ در صف انتظار بوده است.	$time_{hold}$
مدت زمان انجام کار $j$	$time_j$
مدت زمان پاسخگویی به تمامی درخواست‌های ممکن کاربران	$time_{all}$
توان مصرفی پردازنده	$P_i$
سطح فرکانسی پردازنده در لحظه $t$	$f(t)$
مصرف انرژی پردازنده	$E_i$
انرژی مصرف شده	$Energy Consumption$

نظر گرفته شد، بدین شرح می‌باشد: تعداد آزمایش هشت، تعداد تکرار الگوریتم ۵۰، تعداد ماشین‌های مجازی نه، تعداد وظایف ده و تعداد میزبان فیزیکی دو. میانگین نتایج حاصل از انجام آزمایشات در جداول ۲، ۳ و ۴ گزارش شده است.

جدول ۲: سود (بر حسب دلار)

Run	WBA	BA	ACO	ICA	GA
۱	۲۰۷۱۱۳،۷۰	۲۰۶۴۳۴،۹۹	۲۰۵۳۶۸،۲۰	۲۰۶۲۳۱،۶۵	۲۰۱۶۵۸،۳۹
۲	۲۰۷۵۶۸،۹۱	۲۰۶۱۹۳،۸۵	۲۰۱۳۹۶،۸	۲۰۳۴۸۲،۳۹	۲۰۲۵۴۸،۵۲
۳	۲۰۷۲۲۰،۳۳	۲۰۶۱۴۶،۵۴	۲۰۴۶۳۵،۶۵	۲۰۵۶۲۴،۸۶	۲۰۳۶۵۸،۲۱
۴	۲۰۷۰۶۲،۳۹	۲۰۶۸۱۴،۰۳	۲۰۳۲۵۶،۳۵	۲۰۶۵۲۱،۳۶	۲۰۱۴۵۷،۳۶
۵	۲۰۷۶۱۳	۲۰۶۸۲۸،۴۶	۲۰۶۵۴۲،۶۹	۲۰۶۳۲۵،۵۴	۲۰۲۵۹۸،۳۶
۶	۲۰۷۱۱۵،۳۰	۲۰۶۵۰۱،۵۳	۲۰۴۵۲۳،۹	۲۰۵۲۴۵،۱۸	۲۰۳۵۴۸،۶۷
۷	۲۰۷۵۳۴،۴۳	۲۰۶۱۲۸،۹۱	۲۰۵۳۶۸،۳۱	۲۰۵۲۱۴،۲۶	۲۰۲۹۸۵،۳۲
۸	۲۰۷۲۱۳،۱۱	۲۰۶۵۷۵،۳۹	۲۰۴۹۸۷،۳۶	۲۰۵۳۶۹،۹۸	۲۰۴۳۶۹،۲۸

جدول ۳: زمان پاسخگویی (بر حسب ثانیه)

Run	WBA	BA	ACO	ICA	GA
۱	۴۱۹۲،۶۰	۱۹۵۶۷،۳۰	۲۱۳۵۶،۲	۲۰۳۲۵،۶۰	۲۲۳۶۵
۲	۳۷۱۷	۱۶۸۷۲،۳۰	۲۰۲۵۴،۳۰	۱۹۶۲۵،۳۰	۲۳۹۶۵
۳	۳۵۶۷،۳۰	۱۴۹۲۷،۳۰	۲۱۹۶۸	۱۸۲۵۶،۶۰	۲۱۶۳۵،۳۰
۴	۱۹۴۷،۶۰	۴۹۸۲،۶۰	۱۸۳۶۹	۱۸۹۶۵،۳۰	۲۲۱۵۶
۵	۲۶۴۷،۳۰	۱۴۹۱۲،۶۰	۱۹۶۳۵	۱۶۵۲۶،۳۰	۲۱۵۷۸،۶۰
۶	۳۶۸۷	۳۳۳۷،۶۰	۲۱۹۶۵	۱۹۳۲۶	۲۰۵۶۹
۷	۲۴۵۷،۳۰	۵۰۸۲،۳۰	۱۶۵۲۳	۱۶۳۲۹	۲۱۵۷۴
۸	۲۳۱۸،۳۰	۱۵۷۳۷،۶۰	۱۸۹۲۵	۱۹۲۵۶	۲۲۳۲۴،۳۰

جدول ۴: انرژی صرف شده (بر حسب کیلووات بر ساعت)

Run	WBA	BA	ACO	ICA	GA
۱	۲،۱۱	۴،۵۷	۴،۶۱	۴،۲۱	۵،۱
۲	۲،۲۴	۴،۳۵	۴،۵۲	۴،۳۶	۴،۹
۳	۲،۱۸	۴،۱۷	۴،۳۹	۴،۳۹	۴،۸۱
۴	۲،۰۴	۳،۹۴	۴،۵۱	۴،۱۹	۴،۶۱
۵	۱،۹۰	۲،۶۴	۴،۵۶	۴،۲۱	۴،۹
۶	۲،۲۷	۲،۴۹	۴،۰۱	۴،۱۰	۴،۳۹
۷	۱،۹۱	۲،۷۹	۴،۹	۴،۲۱	۴،۵۱
۸	۲،۰۷	۴،۰۲	۴،۵	۴،۳۶	۴،۹

نمودارهای مقایسه الگوریتم‌های آزمایش شده در هر سه تابع معیار در شکل ۲ آورده شده است.

#### ۴- شبیه سازی و ارزیابی

به منظور ارزیابی عملکرد مدل ارائه شده با الگوریتم خفاش و الگوریتم خفاش پیشنهادی، ما مسئله را با پارامترهای جدول یک در نظر گرفتیم و نتایج آزمایشات را با الگوریتم ژنتیک [21]، مورچگان [22] و رقابت استعماری [23] مقایسه نمودیم. شبیه سازی در محیط نرم افزار NetBeans 8.2 و شبیه ساز قدرتمند Cloudsim 3.0.3 با پردازشگر Core™ i3 3.70 GHz و Ram 4GB صورت گرفته است. پارامترهای الگوریتم خفاش استاندارد و الگوریتم خفاش پیشنهادی ما برابر با  $\alpha=\gamma=0.9$ ،  $F_{min}=0$ ،  $F_{max}=100$ ،  $V_{min}=1$ ،  $V_{max}=4.5$ ،  $W_{min}=0.36$ ،  $W_{max}=0.96$  و Bat number = 16 در نظر گرفته شد. دیتای آزمایش مشابه دیتای [24] بوده که بسیار نزدیک به دنیای واقعی است. پارامترهای تست که برای همه الگوریتم‌ها به صورت یکسان در

۴	الگوریتم مورچگان	این الگوریتم به علت استفاده از تعداد مورچه بیشتر از زمان زیادی برای اجرا برخوردار بود، اما فضای جستجوی قابل قبولی در اختیار برنامه قرار داد.
۵	الگوریتم ژنتیک	با وجود اینکه پیچیدگی محاسباتی کمتری داشت، اما به علت محدودیت در فضای جستجو خوب عمل نکرد و نتوانست به نتایج خوبی دست یابد.

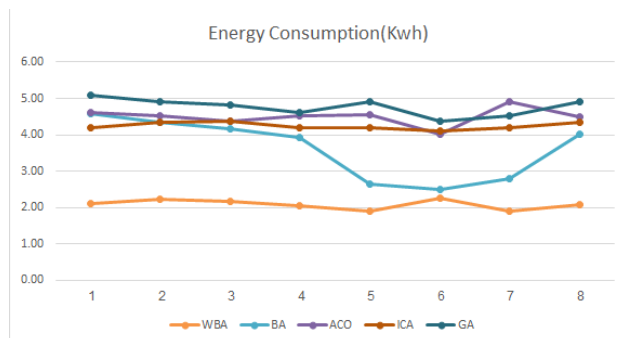
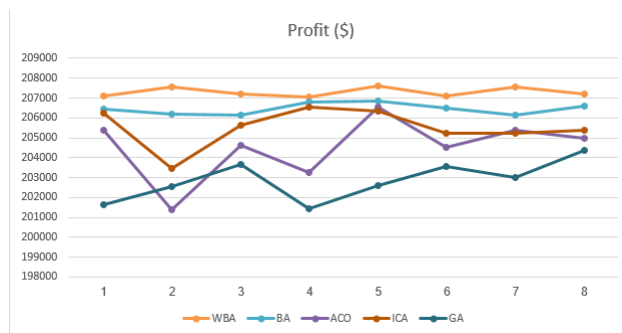
## ۵- نتیجه گیری

پس از شبیه سازی مدل با استفاده از الگوریتم خفاش پیشنهادی برای تخصیص و قیمت گذاری منابع ابری در روش حراج دو طرفه و انجام آزمایشات گسترده جهت ارزیابی مدل، نتایج آزمایشگاهی حاکی از برتری مدل ما در همه موارد نسبت به سایر الگوریتم های فراابتکاری مورد مقایسه ژنتیک، مورچگان و رقابت استعماری می باشد. ما مدلمان را با سه تابع هدف سود ابر، زمان محاسباتی و انرژی صرف شده ارزیابی و با سایر روش ها مقایسه نمودیم. در تمامی موارد، الگوریتم خفاش پیشنهادی منابع را با هزینه کمتر و در زمان کوتاهتری در اختیار کاربران قرار می دهد. هر محقق در مسیر پژوهش، جمع آوری اطلاعات و کسب نتایج مطلوب با مشکلاتی روبه رو می شود. یکی از محدودیت های تحقیق، نبود داده های واقعی از بازار حراج دوطرفه است که ما را مجبور به استفاده از دیتای شبیه سازی شده نموده است.

بدیهی است یکی از کارهای آینده می تواند توسعه الگوریتم خفاش ارائه شده به صورت چند هدفه و لحاظ همه گزینه های موجود در بازار دوطرفه باشد. هم چنین از آنجا که مکانیزم بازار حراج دو طرفه دارای نیازمندی های پیچیده ای است و بستر اجرای آن در دنیای واقعی تاکنون آماده نگردیده است، لذا انجام تحقیقات با جزئیات کامل بسترسازی این بازار جهت مبادلات خدمات ابری و استقرار نظام پیشنهادی حراج نیازمند بررسی های عمیق تری است.

## مراجع

- [1] X. Zhang, Z. Huang, C. Wu, Z. Li and F. Lau, "Online Auctions in IaaS Clouds: Welfare and Profit Maximization with Server Costs," *ACM SIGMETRICS '15 Proceedings*, pp. 3-15, 2015.
- [2] S. Chichin, Q. Bao Vo and R. Kowalczyk, "Double-sided Market Mechanism for Trading Cloud Resources," In the Proceedings of the 2014 IEEE/WIC/ACM International Joint Conferences on WebIntelligence(WI) and IntelligentAgentTech- nologies (IAT), Warsaw, Poland, 11-14 August 2014, 2015, pp. 198-205.
- [3] M. A. Adibi and J. Shahrabi, " Improving the performance of the variable neighborhood search algorithm with clustering," the 9<sup>th</sup> Conference of Data Mining Dec. 20&21, 2015, Tehran, Iran.



شکل ۲: نمودار مقایسه توابع معیار در الگوریتم های آزمایش شده.

بر اساس نتایج حاصله، الگوریتم خفاش وزن دار پیشنهادی در تمامی موارد بهتر از سایر الگوریتم های مقایسه شده عمل نموده است. جدول ۵ اولویت بندی روش های پیشنهادی را طبق نتایج حاصله از آزمایشات بر اساس قدرت و مطلوبیت نشان می دهد.

جدول ۵: جدول اولویت بندی روش های پیشنهادی

ردیف	روش پیشنهادی	مزیت ها و معایب
۱	الگوریتم خفاش پیشنهادی	استفاده از فضای جستجوی بزرگ تر و پیچیدگی محاسباتی کمتر و ثبات در نتایج حاصل از آزمایش های مختلف.
۲	الگوریتم خفاش	استفاده از فضای جستجوی بزرگ تر و پیچیدگی محاسباتی کمتر
۳	الگوریتم رقابت استعماری	جستجو در فضای بزرگ تر و استفاده از دو شرط برای خاتمه الگوریتم.



- [4] V. Kamra, K. Sonawane and P. Alappanavar, "Cloud Computing And Its Pricing Schemes," *International Journal on Computer Science and Engineering (IJCSE)*, Vol. 4, No. 4, pp. 577-581, 2012.
- [5] X. Wu, M. Wang, Sh. Wu Zhang and Y. Guo, "Cloud Program with a Pricing Strategy for Iaas in Cloud Computing," in *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW)*, *IEEE 26<sup>th</sup> International*, 2012, pp. 2316-2319.
- [6] X. Hong and L. Baochun, "A Study of Pricing for Cloud Resources," *ACM Performance Evaluation Review*, Vol. 40, No. 4, pp. 3-12, 2013.
- [7] M. Mihailescu and Y. Teo, "Dynamic Resource Pricing on Federated Clouds," *10<sup>th</sup> IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, 2010, pp. 513-517.
- [8] Q. Wang, K. Ren and X. Meng, "When Cloud Meets eBay: Towards Effective Pricing for Cloud Computing," *IEEE INFOCOM*, pp. 936-944, 2012.
- [9] L. Hao, L. Jianhui and T. Guo, "A Pricing Algorithm for Cloud Computing Resources," *International Conference on Network Computing and Information Security (NCIS)*, 2011, pp. 69-73.
- [10] A. Ojala, "Selection of the Proper Revenue and Pricing Model for SaaS," *IEEE 6<sup>th</sup> International Conference on Cloud Computing Technology and Science*, 2014, pp. 863-868.
- [11] G. Laatikainen and A. Ojala, "SaaS architecture and pricing models," *IEEE international conference (SCC)*, 2014, pp. 597-604.
- [12] M. Mac'ias and J. Guitart, "A Genetic Model for Pricing in Cloud Computing," in *ACM Symposium on Applied Computing*, 2011, pp. 113-118.
- [13] W. Wang, B. Liang and B. Li, "Revenue Maximization with Dynamic Auctions in IaaS Cloud Markets," in *IEEE/ACM 21<sup>st</sup> (IWQoS)*, 2013, pp. 1-6.
- [14] A. Nadjaran, K. Mechelen and R. Buyya, "An Auction Mechanism for a Cloud Spot Market," *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, Vol. 11, No. 2, 2016.
- [15] N. Fukuta and T. Ito, "Fine-grained efficient resource allocation using approximated combinatorial auctions," *IJHDS*, 2009, pp. 43-63.
- [16] S. Chichin, Q. Bao Vo and R. Kowalczyk, "Truthful Market-based Trading of Cloud Services with Reservation Price," in *the Proceedings of the 11<sup>th</sup> IEEE International Conference on Services Computing (SCC 2014)*, Anchorage, Alaska, USA, June 27 - July 2, 2014, pp. 27-34.
- [17] S. Chichin, Q. Bao Vo and R. Kowalczyk, "Adaptive Market Mechanism for Efficient Services Trading," In *the Proceedings of the 7<sup>th</sup> IEEE International Conference on Cloud Computing (CLOUD 2014)*, Anchorage, Alaska, USA, June 27 - July 2, 2014, pp. 705-712.
- [18] S. Chichin, Q. Bao Vo and R. Kowalczyk, "Towards Efficient Greedy Allocation Schemes for Double-sided Cloud Markets," In *the Proceedings of the 12<sup>th</sup> IEEE International Conference on Services Computing (SCC 2015)*, New York, USA, June 27 - July 2, 2015, pp. 194-201.
- [19] X. S. Yang, "A new metaheuristic bat-inspired algorithm" in: J. R. Gonzalez editor, *Nature inspired cooperative strategies for optimization (NICSO 2010)*, Springer Press., vol. 284, pp. 65-74, 2010.
- [20] X. S. Yang, "Bat algorithm for multi-objective optimization," *International Journal Bio-Inspired Comput* 3(5), pp. 74-267, 2011.
- [21] B. Madhusudhan and Ch. Sekaran, "a genetic algorithm approach for virtual machine placement in cloud," in the proceeding of the Elsevier international conference on Emerging research in computing, information communication and applications, 2013, pp. 115-122.
- [22] E. Pacini, C. Mateos and C. G. Garino, "Balancing throughput and response time in online scientific Clouds via Ant Colony Optimization," *Journal of Elsevier, Advances in engineering software*, pp. 1-17, 2015.
- [23] R. Asemi, E. Doostsadiq, M. Ahmadi and H. Tabatabaee Malazi, "Energy Efficiency in Virtual Machines Allocation for Cloud Data Centers using the Imperialist Competitive Algorithm," *The proceeding of 5<sup>th</sup> IEEE International Conference on Big data and cloud computing (BDCloud)*, china, Aug 26-28, 2015.
- [24] S. Chichin, "An Open Market for Trading Cloud Services", Ph.D. dissertation, Dept. Eng.Tech., Swinburne univ., Australia, 2016.