

## زمان بندی تطبیقی سیستم های بی درنگ تغذیه شونده با انرژی های تجدید پذیر

مریم دهقان منشادی<sup>۱</sup>

مهدی کارگهی<sup>۱و۲</sup>

<sup>۱</sup>دانشکده مهندسی برق و کامپیوتر، پردیس دانشکده های فنی، دانشگاه تهران، تهران، ایران  
<sup>۲</sup>پژوهشکده علوم کامپیوتر، پژوهشگاه دانش های بنیادی (IPM)، تهران، ایران

### چکیده

به دلیل ماهیت سیستم های بی درنگ نهفته، تامین و مدیریت انرژی در بسیاری از این سیستم ها یک موضوع پراهمیت و یک چالش جدی محسوب می گردد. از این رو، در مطالعات اخیر از برداشت انرژی بعنوان یک راهکار مفید برای تامین انرژی آن ها استفاده شده است. لیکن، طبیعت تصادفی دریافت انرژی های تجدید پذیر، مدیریت انرژی و عملکرد چنین سیستم هایی را با مشکل مواجه می کند. در این مقاله به ارائه دو الگوریتم زمان بندی برخط می پردازیم که از امکان تغییر پویای ولتاژ در پردازنده های امروزی برای حل این مشکل بهره می گیرند: الگوریتم EAS<sup>o</sup> برای حالتی که ظرفیت مخزن انرژی نامحدود باشد و الگوریتم EAS<sup>c</sup> برای حالتی که این ظرفیت محدود باشد. برای شرایطی که امکان پیش بینی دقیق انرژی برداشتی از محیط مهیا باشد، الگوریتم اول بهینه است که این موضوع اثبات می گردد. در مورد الگوریتم دوم نیز با استفاده از شبیه سازی نشان داده می شود که می توان با اجتناب از سرریز انرژی مخزن تا جای ممکن هدررفت انرژی را کاهش داده و کارایی سیستم را به نحو قابل ملاحظه ای بهبود داد. کارآمدی الگوریتم های پیشنهادی در مقایسه با EDF، یک الگوریتم پایه مبتنی بر DVFS به نام staticEDF، و یک الگوریتم مبتنی بر DVFS برای سیستم های برداشت گر انرژی به نام EA-DVFS مورد ارزیابی قرار می گیرد.

**کلمات کلیدی:** برداشت انرژی، تغییر پویای ولتاژ، زمان بندی بی درنگ، سیستم های بی درنگ، سیستم های نهفته.

### ۱- مقدمه

غیرممکن باشد این محدودیت جدی تر می شود. امکان برداشت انرژی از محیط یکی از راهکارهای نوین و موثر برای حل این معضل محسوب می گردد [۱]، [۲]، [۳]، [۴].

در [۵] چندین فن آوری مرتبط با برداشت انرژی های محیطی نظیر انرژی خورشیدی، حرارتی و لرزشی مورد بحث قرار گرفته اند. لیکن، به دلیل طبیعت غیرقطعی و تصادفی الگوی دریافت انرژی از این منابع، تضمین سرویس های بی درنگ در این سیستم ها با مشکل مواجه می گردد. به خصوص در سیستم های بی درنگ سخت (از طبقه بندی کلاسیک سیستم های بی درنگ) که آماده شدن دیرنگام نتایج تبعات فاجعه آمیز دارد استفاده از انرژی های تجدید پذیر با طبیعت غیرقطعی مذکور غیرقابل پذیرش خواهد بود. در مقابل، در سیستم های بی درنگ نرم و محکم که تعداد محدود نقض محدودیت زمانی (موعد) فقط باعث کاهش کارایی سیستم شده و تبعات بحرانی جدی ندارد، استفاده از این منابع انرژی معقول خواهد بود.

بنا به پذیرفته شده ترین تعاریف، مشخصه های سیستم هایی بی درنگ نهفته عبارتند از: خاص منظوره بودن، دارا بودن تعامل با محیط پیرامون بواسطه ی حسگرها و عمل گرها، و دارا بودن محدودیت زمانی برای عملیات مورد نظر سیستم. در سال های اخیر، رویه ی استفاده از سیستم های بی درنگ و نهفته در کاربردهای متعدد نظیر سیستم های مخابراتی، ناوبری، جمع آوری اطلاعات، نظامی، و غیره به نحو چشمگیری رو به افزایش بوده است. با توجه به چالش تامین انرژی در بیشتر این سیستم ها و محدودیت مخزن انرژی (مثلا باتری) در بسیاری از آن ها، طول عمر سیستم (متأثر از طول عمر باتری) به یک محدودیت جدی تبدیل شده است. به خصوص در مواردی که تعویض مخزن انرژی یا شارژ مجدد آن به دلایلی نظیر محل استفاده از دستگاه های مربوطه یا متحرک بودن آنها بسیار سخت یا حتی

همان‌گونه که در بخش ۱ نیز اشاره گردید، بیشتر مطالعاتی که از DVFS بهره می‌برند با هدف کمینه کردن انرژی مصرفی ارائه شده‌اند. به‌طور مرسوم این الگوریتم‌ها به سیستم‌هایی اعمال شده‌اند که از محیط انرژی برداشت نمی‌کرده‌اند بلکه از منابع محدود انرژی نظیر باتری استفاده می‌کرده‌اند. این در حالی است که در سیستم‌های آگاه از انرژی که از محیط انرژی دریافت می‌کنند، هدف، استفاده‌ی بهینه از انرژی برداشتی است که ممکن است لزوماً کمینه کردن انرژی برداشتی نباشد [۱۱]، [۱۳]، [۱۴]. به‌عنوان مثال، در شرایطی که انرژی مصرفی کمینه گردد ممکن است به دلیل دریافت انرژی از محیط مخزن انرژی پر شود و دیگر امکان ذخیره‌ی انرژی برداشتی از محیط وجود نداشته باشد. حال آن‌که در صورت استفاده‌ی بهینه از انرژی ممکن است بتوان با اجتناب از صرفه‌جویی، بیشترین انرژی محیط را مورد استفاده قرار داد و از هدر رفتن آن جلوگیری نمود. بر این اساس می‌توان شرایطی را به وجود آورد که در موارد کمتری با کمبود انرژی و مختل شدن عملکرد سیستم مواجه شویم.

به‌عنوان نمونه‌ای از الگوریتم‌های مبتنی بر DVFS ولی ناآگاه از انرژی برداشتی می‌توان به [۶] اشاره نمود که الگوریتم RT-DVS را برای کاهش انرژی مصرفی در رده‌ی خاصی از سیستم‌های بی‌درنگ ارائه می‌کند. در یک مطالعه‌ی دیگر [۱۵]، یک الگوریتم زمان بندی بهینه مبتنی بر DVFS به نام LPEDF برای مجموعه‌ای از کارها با زمان‌های ترخیص و موعد دلخواه ارائه شده است که به‌صورت برون خط و بر مبنای الگوریتم EDF برای وظایف مستقل عمل می‌نماید. در [۱۶] نیز یک الگوریتم تعمیم‌یافته برای کلیه‌ی وظایف دوره‌ای، غیردوره‌ای و پراکنده ارائه شده است که تعمیمی از چندین الگوریتم معروف مبتنی بر DVFS محسوب می‌گردد.

بر خلاف گستردگی مطالعات در زمینه بهینه‌سازی مصرف انرژی، کارهای انجام شده در زمینه الگوریتم‌های زمان بندی بی‌درنگ سیستم‌های برداشت‌گر انرژی محدود هستند. یکی از پایه‌ای‌ترین کارهای انجام شده در این زمینه [۱۰] است که به ارائه‌ی یک الگوریتم بهینه به نام Lazy برای زمان بندی بی‌درنگ برخط شبکه‌های حسگر بی‌سیم می‌پردازد. مدل کارها در این الگوریتم بگونه‌ای است که هر کار باید میزان مشخصی از انرژی سیستم را در محدوده‌ی زمانی مشخص به دست آورد که به همین دلیل اعمال DVFS روی آن برای کاهش مصرف انرژی غیرممکن است. ایده‌ی دیگری در [۱۷] ارائه شده است که برای وظایف دوره‌ای با زمان اجرا و میزان پاداش مشخص در سیستمی که از انرژی خورشیدی استفاده می‌کند یک الگوریتم تغییر پویای ولتاژ ارائه می‌نماید. این الگوریتم بر اساس برخی تصمیم‌های برون خط پاداش سیستم را بهینه می‌کند. در [۱۸] نیز ایده‌ی مشابهی برای وظایف دوره‌ای با موعد یکسان و با فرض ثابت بودن انرژی برداشتی از محیط ارائه شده است. در [۱۹] نیز یک الگوریتم مدیریت پویای توان به نام LM-APM ارائه شده است که در زمان‌های پر بودن باتری از DVFS برای تطبیق بار با نقطه‌ی اوج انرژی برداشتی از محیط استفاده می‌کند و به این ترتیب باعث افزایش زمان لختی می‌گردد.

در الگوریتم ارائه شده در [۲۰]، محدودیت‌های زمانی و انرژی به‌صورت جداگانه منظور می‌شوند و با استفاده از پیش‌بینی کوتاه‌مدت سرریز مخزن انرژی از هدررفت آن اجتناب می‌گردد. در [۷] نیز الگوریتمی به نام EA-DVFS ارائه شده است که با استفاده‌ی کارآمد از لختی و وظیفه‌ی میزان از دست رفتن موعدها را کاهش می‌دهد. در مواردی که مخزن انرژی پر باشد این الگوریتم وظایف را با حداکثر سرعت اجرا می‌کند و در غیر این صورت با استفاده از DVFS در مصرف انرژی صرفه‌جویی می‌نماید.

در الگوریتم‌های ارائه شده در مقاله‌ی حاضر محدودیت‌های زمان و انرژی به‌صورت توأم منظور می‌شوند. بر خلاف مدل وظایف ارائه شده در [۱۰] که از حداکثر انرژی مورد نیاز هر وظیفه آگاه بود، در مطالعه‌ی حاضر، از حداکثر زمان اجرای هر وظیفه آگاه خواهیم بود. لذا، با این تغییر نسبت به مدل وظایف ارائه

یکی از زمینه‌های تحقیقاتی مرتبط که در مطالعات اخیر به آن توجه زیادی شده است ارائه‌ی الگوریتم‌های زمان بندی آگاه از انرژی برداشتی بوده است. بنا به تعریف، صفت آگاه از انرژی برداشتی برای الگوریتم‌های زمان بندی بکار برده می‌شود که تخمینی از انرژی برداشتی سیستم را می‌دانند و بر اساس آن تصمیمات خود را اتخاذ می‌کنند. یکی از روش‌های مرسوم که چنین الگوریتم‌هایی برای استفاده‌ی کارآمد از انرژی برداشتی از آن استفاده می‌کنند تغییر پویای ولتاژ و فرکانس پردازنده با استفاده از مکانیزم قدرتمند DVFS است که کاربرد جدی در مدیریت انرژی دارد.

در بیشتر مطالعات گذشته، DVFS با هدف کاهش انرژی مصرفی سیستم مورد استفاده قرار گرفته است. حال آنکه در سیستم‌های برداشت‌گر انرژی که به منابع نامحدود ولی غیرقطعی انرژی دسترسی دارند هدف اصلی کاهش انرژی مصرفی نیست، بلکه تطابق انرژی مصرفی با انرژی قابل برداشت از محیط با لحاظ نمودن معیارهای کارایی سیستم است. در این راستا می‌توان بجای استفاده از DVFS برای صرفه‌جویی در انرژی، از این مکانیزم برای دستیابی به تطابق مذکور بهره برداری نمود.

در مقاله‌ی حاضر به ارائه‌ی دو الگوریتم زمان بندی برخط برای سیستم‌های برداشت‌گر انرژی‌های تجدیدپذیر با وظایف دوره‌ای بی‌درنگ محکم می‌پردازیم. الگوریتم‌های ارائه شده از DVFS برای تطبیق انرژی مصرفی با الگوی دریافت انرژی استفاده می‌کنند. دو مدل مخزن با ظرفیت محدود و مخزن با ظرفیت نامحدود در نظر گرفته شده اند که مدل دوم معادل حالتی است که مخزن انرژی بسیار بزرگ تر از میزان انرژی قابل برداشت باشد. دو الگوریتم ارائه شده متناسب با این دو ظرفیت مخزن طراحی شده‌اند. شبیه‌سازی‌های انجام شده عملکرد بهتر الگوریتم‌های پیشنهادی نسبت به الگوریتم‌های EDF، staticEDF [۶] و EA-DVFS [۷] را تایید می‌کند.

ساختار ادامه این مقاله به شرح زیر است. در بخش ۲، توضیحاتی در رابطه با کارهای مرتبط با پژوهش حاضر ارائه می‌گردد. در بخش ۳، مدل سیستم، مشتمل بر مدل‌های وظایف، پردازنده، مخزن انرژی، و دریافت و مصرف انرژی ارائه شده‌اند. بخش ۴ ایده‌ی راه‌حل ارائه شده و الگوریتم‌های زمان بندی متناظر با هر دو مدل باتری را ارائه می‌نماید. یک الگوریتم تکمیلی در بخش ۵ ارائه شده است. نتایج تجربی و جمع‌بندی مقاله هم به ترتیب در بخش‌های ۶ و ۷ ارائه شده‌اند.

## ۲- کارهای مرتبط

به دلیل طبیعت تصادفی میزان انرژی قابل برداشت در سیستم‌های برداشت‌گر انرژی، متداول است که برای کاهش میزان نقض موعد، از برخی مدل‌های انرژی و الگوریتم‌های پیش‌بینی نیز در کنار سیاست‌های مناسب زمان بندی وظایف استفاده گردد. برای مثال، در [۸] الگوریتمی برای پیش‌بینی انرژی خورشیدی ورودی به سیستم بر اساس تغییر فصول سال و همین‌طور تغییر ساعات طلوع و غروب خورشید ارائه شده است. در [۹] یک زنجیره مارکوف زمان-گسسته بر اساس یک سال مشاهدات مربوط استخراج شده است که برای پیش‌بینی انرژی برداشتی مورد استفاده قرار می‌گیرد.

در [۱۰] نیز مشخصات یک منبع انرژی بر اساس برخی منحنی‌های تعیین تغییرات انرژی ارائه شده است که میزان انرژی برداشتی در یک حوزه زمانی را در قالب یک کران بالا یا یک کران پایین ارائه می‌نماید. در [۱۱] یک مدل تحلیلی برای تنظیم چرخه کار سیستم بر اساس مشخصه‌های منبع انرژی خورشیدی ارائه شده است. در [۱۲] هم الگوریتم‌های پیش‌بینی انرژی خورشیدی مربوط به سیستم‌های برداشت‌گر انرژی از جنبه‌های دقت، سربار زمانی و سربار حافظه با یکدیگر مقایسه شده‌اند.

هر وظیفه دنباله‌ای مستقل از کارها است که در مرز دوره‌ی هر وظیفه ترخیص می‌شوند. [امین کار وظیفه‌ی  $T_i$  که توسط نماد  $J_{i,j}$  نمایش داده می‌شود دارای مشخصات زیر است:

- $J_{i,j}$  شده و به وسیله‌ی سه تایی مرتب  $(r_{i,j}, R_{ex_{i,j}}, R_{D_{i,j}})$  توصیف می‌شود که  $r_{i,j}$  زمان ترخیص کار،  $R_{ex_{i,j}}$  تعداد سیکل‌های اجرایی باقی‌مانده، و  $R_{D_{i,j}}$  موعد نسبی باقی‌مانده است،
  - از آنجا که وظایف دوره‌ای هستند،  $r_{i,j} = (j - 1)P_i$  خواهد بود،
  - $R_{ex_{i,j}}$  در ابتدا مقدار  $C_i$  را داراست و پس از اجرای  $c$  سیکل از آن کار به  $C_i - c$  تغییر مقدار می‌دهد،
  - $R_{D_{i,j}}$  هم در ابتدا مقدار  $D_i$  گرفته و با گذشت  $t$  واحد زمانی به  $D_i - t$  تغییر می‌کند.
- زمان بندی برای یک ابردوره (کوچکترین مضرب مشترک دوره‌های وظایف) که با  $H$  نمایش داده می‌شود تعیین می‌گردد.

### ۳-۲- مدل پردازنده

وظایف سیستم روی یک پردازنده‌ی مجهز به DVFS با قابلیت فعالیت در محدوده‌ی فرکانسی  $[f_{min}, f_{max}]$  اجرا می‌شوند. تابع  $Pow(f)$  توان مصرفی پردازنده در فرکانس  $f$  در محدوده‌ی مذکور و  $Pow(f) \times t$  میزان انرژی مصرفی در دوره‌ای به طول  $t$  را نشان می‌دهد. هر دو سربار مقیاس کردن ولتاژ پردازنده و تعویض متن در زمان اجرای وظیفه منظور شده‌اند. در ضمن، حداکثر فرکانس پردازنده به مقدار ۱ نرمال شده است.

### ۳-۳- مدل مخزن انرژی

مخازن انرژی ممکن است باتری یا ابرخازن باشند. دو نوع مخزن، یکی با ظرفیت محدود  $C$  و دیگری با ظرفیت نامحدود در نظر گرفته شده‌اند. البته مخزن با ظرفیت نامحدود ایده‌آل است و فقط جنبه نظری دارد. در اینجا مقصود مخزنی است که ظرفیت آن در مقابل انرژی برداشتی از محیط بسیار بزرگ باشد. نماد  $e_b(t)$  در ادامه‌ی مقاله بیانگر میزان انرژی ذخیره‌شده در مخزن در لحظه‌ی  $t$  خواهد بود.

### ۳-۴- مدل‌های دریافت و مصرف انرژی

با توجه به تعریف  $P_h(t)$  که پیش‌تر ذکر شد، انرژی برداشت شده  $e_h$  در بازه‌ی زمانی  $[t_1, t_2]$  به صورت زیر محاسبه می‌شود (اگر  $t_1$  زمان جاری و  $t_2$  زمانی در آینده باشد،  $e_h(t_1, t_2)$  بیانگر پیش‌بینی ما از انرژی برداشتی در این بازه‌ی زمانی خواهد بود):

$$e_h(t_1, t_2) = \int_{t_1}^{t_2} P_h(t) dt \quad (1)$$

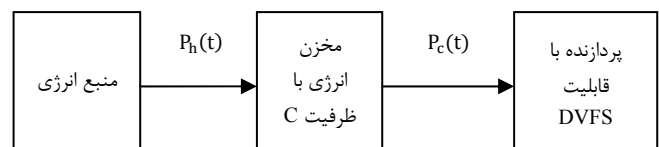
به طور مشابه و بنا به تعریف  $P_c(t)$  که در بالا ارائه شد، انرژی مصرف شده از مخزن در بازه‌ی زمانی  $[t_1, t_2]$  از رابطه‌ی  $e_c(t_1, t_2) = \int_{t_1}^{t_2} P_c(t) dt$  پیروی کرده و لذا نامساوی زیر برقرار خواهد بود:

$$e_c(t_1, t_2) \leq e_b(t_1) + e_h(t_1, t_2) \quad (2)$$

شده در [۱۰]، امکان استفاده از DVFS برای تطبیق انرژی مصرفی با الگوی دریافت انرژی فراهم گشته و مورد استفاده قرار می‌گیرد.

### ۳- مدل سیستم

در این بخش به معرفی اجزای تشکیل دهنده‌ی سیستم مورد بحث در این مقاله می‌پردازیم. همان گونه که در شکل ۱ نیز دیده می‌شود، انرژی برداشتی از محیط در لحظه‌ی  $t$  به انرژی الکتریکی با نرخ  $P_h(t)$  تبدیل می‌شود که در یک مخزن انرژی ذخیره می‌گردد. از طرف دیگر، یک دستگاه محاسباتی با نرخ  $P_c(t)$  انرژی این مخزن را مصرف نموده و وظایف سیستم را توسط یک پردازنده با قابلیت DVFS اجرا می‌نماید. توضیحات دقیق‌تر اجزای این سیستم در ادامه ارائه می‌گردند. در ضمن، جدول ۱ نمادگذاری استفاده شده در این بخش را نشان می‌دهد.



شکل ۱- سیستم برداشت‌گر انرژی

### ۳-۱- مدل وظیفه

مجموعه‌ای از  $n$  وظیفه‌ی بی درنگ  $T = \{T_1, T_2, \dots, T_n\}$  با مشخصات و فرضیات زیر در سیستم موجود هستند:

- هر وظیفه‌ی  $T_i$  دوره‌ای و غیرانحصاری است و توسط یک تاپل  $(P_i, C_i, D_i)$  معرفی می‌گردد،
- $P_i$  بیانگر دوره وظیفه و  $D_i$  بیانگر موعد نسبی است و برای تمام وظایف رابطه‌ی  $P_i = D_i$  برقرار است،
- $C_i$  حداکثر تعداد سیکل مورد نیاز برای اجرای وظیفه‌ی  $T_i$  است.

جدول ۱- نمادگذاری

n	تعداد وظایف سیستم	f	فرکانس پردازنده
$T_i$	وظیفه $T_i$	$f_{min}$	حداقل فرکانس پردازنده
$P_i$	دوره وظیفه $T_i$	$f_{max}$	حداکثر فرکانس پردازنده
$C_i$	زمان اجرای وظیفه $T_i$	C	ظرفیت مخزن انرژی
$D_i$	موعد نسبی وظیفه $T_i$	$e_b(t)$	انرژی مخزن در لحظه t
$J_{i,j}$	کار لازم از وظیفه $T_i$	$P_h(t)$	توان قابل برداشت در لحظه t
$r_{i,j}$	زمان ترخیص $J_{i,j}$	$e_h(t_1, t_2)$	انرژی قابل برداشت در بازه $[t_1, t_2]$
$R_{ex_{i,j}}$	زمان اجرای باقی مانده از $J_{i,j}$	$P_c(t)$	توان مصرفی در لحظه t
$R_{D_{i,j}}$	موعد نسبی باقی مانده از $J_{i,j}$	$e_c(t_1, t_2)$	انرژی مصرفی در بازه $[t_1, t_2]$
H	ابردوره	L	طول برش
		slot <sub>i</sub>	برش نام

## ۴- الگوریتم EAS برای زمان بندی آگاه از انرژی برداشتی

در این بخش به ارائه ای الگوریتم های پیشنهادی مقاله حاضر می پردازیم.

### ۴-۱- کلیات راه حل پیشنهادی

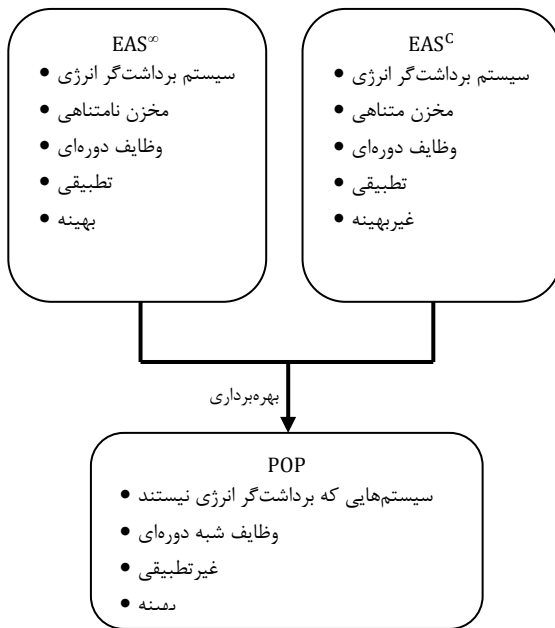
در این بخش به ارائه ای کلیات راه حل پیشنهادی برای مساله ای زمان بندی وظایف در یک سیستم بی درنگ برداشت گر انرژی می پردازیم. دو الگوریتم زمان بندی تطبیقی به نام های  $EAS^{\infty}$  و  $EAS^C$  برای سیستم های بی درنگ برداشت گر انرژی طراحی شده اند.  $EAS^{\infty}$  یک الگوریتم بهینه برای سیستمی با وظایف دوره ای و ظرفیت نامحدود مخزن است.  $EAS^C$  هم تعمیمی روی الگوریتم قبلی است که برای سیستم های با ظرفیت محدود مخزن طراحی شده است.

الگوریتم اول به دلیل داشتن مخزن بزرگتر سربرار زمانی کمتر دارد. سیستم های غیرمتحرک که قابلیت تامین هزینه مخزن بزرگ برای آنها فراهم باشد می توانند از مزایای این روش بهره ببرند. در حالی که سیستم های متحرک که فضای کافی برای تامین مخزن بزرگ انرژی ندارند، به رغم سربرار زمانی بالاتر، از الگوریتم دوم بهره می گیرند. همانگونه که در شکل ۲ هم نشان داده شده است، هر دو الگوریتم مطرح شده از الگوریتم POP<sup>۱</sup> برای به دست آوردن فرکانس بهینه استفاده می کنند. POP که در این مقاله ارائه شده است یک الگوریتم بهینه ای آگاه از توان برای زمان بندی وظایف شبه دوره ای در سیستم هایی است که از انرژی محیط استفاده نمی کنند. لیکن، بهره برداری از آن به نحوی انجام خواهد شد که برای سیستم های برداشت گر انرژی مناسب باشد. جزئیات وظایف شبه دوره ای و الگوریتم POP در بخش ۴ ارائه خواهد شد.

### ۴-۲- $EAS^{\infty}$ برای مخزن با ظرفیت نامحدود

اگر ظرفیت مخزن انرژی نامحدود فرض شود (لازم به ذکر است که نامحدود بودن را می توان به بسیار بزرگ بودن تعبیر نمود)، هیچگاه باتری سرریز نشده و کل انرژی برداشتی از محیط قابل ذخیره سازی و استفاده در بازه های زمانی بعدی است. برش  $slot_i$  (به عنوان یک بازه ای زمانی به طول  $L_i$  تعریف می شود.  $0 \leq i \leq H/L$ ) از پارامترهای طراحی است که با توجه به  $H$  که بر اساس وظایف سیستم مشخص است تعیین می گردد و ابردوره را تقسیم می کند. به این ترتیب، تعداد برش ها در یک ابردوره  $H/L$  است. طول  $L_i$  باید طوری باشد که احتمال تغییرات پیش بینی انرژی برداشتی در طول آن وجود داشته باشد و بر این اساس در ابتدای هر برش، بنا به تغییرات احتمالی در انرژی برداشتی نسبت به پیش بینی انجام شده، نیاز به بازنگری در زمان بندی ابردوره باشد. کلیه ی مراحل الگوریتم EAS در مرز این برش ها انجام می گردد.

این مراحل شامل محاسبه ی فرکانس پردازنده، بررسی انرژی در دسترس، و برنامه ریزی برای اجرای کارهاست. الگوریتم  $EAS^{\infty}$  در ابتدای هر برش حداقل فرکانس مورد نیاز پردازنده برای اجرای کارهای ابردوره ی جاری را بر اساس باقی مانده ی زمان اجرا و زمان موعدهای کارها محاسبه می کند. سپس، با بررسی میزان انرژی های در دسترس، امکان پذیری بکارگیری این سطح سرعت مورد بررسی قرار می گیرد که در این صورت انتظار می رود تمامی کارها موعده خود را مشاهده نمایند و انرژی مصرفی نیز کمینه گردد. لیکن، به دلیل طبیعت تصادفی انرژی برداشتی از محیط این امکان نیز وجود دارد که در مقاطعی از زمان با کمبود انرژی مواجه شویم.



شکل ۲- نمای کلی راه حل پیشنهادی

بر اساس آنچه ذکر گردید، باید در ابتدای هر برش میزان انرژی قابل دسترس در طول آن برش پیش بینی گردد و بر اساس آن فرکانس پردازنده تعیین گردد. در شرایطی که به نظر برسد انرژی مورد نیاز قابل حصول است کار با زودترین موعد با فرکانس مورد انتظار اجرا خواهد شد. اگر انرژی بیشتری از محیط دریافت شد به سادگی و بدون نگرانی از سرریز انرژی (به دلیل ظرفیت نامحدود مخزن) می توان آنرا برای برش های بعد که احیاناً امکان برداشت کافی انرژی نیست ذخیره نمود. لیکن، در صورت کافی نبودن انرژی در دسترس در هنگام بررسی در مرز برش، کل انرژی در دسترس برای اجرای کار با زودترین موعد مورد استفاده قرار خواهد گرفت. در این شرایط انتظار می رود مخزن انرژی تخلیه شود. در ابتدای هر برش، روند توضیح داده شده بر اساس همین منطق تکرار می گردد.

این الگوریتم برای منابعی همچون خورشید مناسب به نظر می رسد. دلیل این امر این است که انرژی قابل برداشت از خورشید در طول روز زیاد بوده و در شب ناچیز است. لذا یک راه معقول ذخیره ی تمام انرژی اضافه در طول روز برای استفاده در طول شب خواهد بود. الگوریتم  $EAS^{\infty}$  که در ابتدای هر ابردوره اجرا می شود در شکل ۳ نشان داده شده است.

**توضیح الگوریتم:** در ابتدا مجموعه ی کارهای قابل اجرا در طول یک ابردوره و تعداد برش های آن ابردوره (حاصل تقسیم طول ابردوره به طول برش) تعیین می شوند. از خط ۴ رویه ی اصلی الگوریتم زمان بندی آغاز می شود. خط ۸ نقض موعد کارهای مجموعه ی  $J$  را بررسی می کند: به این صورت که اگر موعد نسبی باقی مانده برای یک کار از زمان اجرای باقی مانده ی آن کار کمتر است به معنای آن دست رفتن موعد کار  $J_i$  خواهد بود. خط ۱۰ در صورت اتمام کار با زودترین موعد آن را از مجموعه کارهای سیستم خارج می کند. در خط ۱۲ با استفاده از فراخوانی  $POP(J)$  حداقل فرکانس مورد نیاز پردازنده محاسبه می شود: این فراخوانی بر مبنای باقی مانده ی سیکل های قابل اجرا و باقی مانده ی موعد نسبی کل کارهای ابردوره، در برش جاری فرکانس کمینه را محاسبه می کند. جزئیات بیشتر این تابع در بخش ۴ ارائه می گردد.

خط ۱۳ کار با زودترین موعد را پیدا می کند و خط ۱۴ کمترین فرکانس لازم برای اتمام پیش از موعد آن را محاسبه می نماید. در صورت عدم تامین انرژی مورد نیاز این کار تا قبل از موعد آن، انرژی مصرفی آن کار هدر می رود و لذا بهتر است این کار دور انداخته شود. در ضمن، اگر فرکانس محاسبه شده توسط  $POP(J)$

**قضیه:**  $EAS^\infty$  یک الگوریتم بهینه است، به این معنا که با پیش‌بینی دقیق انرژی برداشت شده و در صورت امکان زمان‌بندی کارها با انرژی در دسترس، به‌طور یقین یک زمان‌بندی ممکن ارائه می‌کند.

**اثبات:** در ابتدا فرض می‌کنیم که الگوریتم POP یک الگوریتم بهینه است که این امر در بخش ۴ اثبات می‌گردد. لذا، در هر بازه‌ی زمانی می‌توان با بهره‌گیری از الگوریتم POP حداقل انرژی مورد نیاز برای مشاهده‌ی کلیه‌ی موعدهای ممکن را محاسبه نمود. از آنجا که ظرفیت مخزن انرژی نامحدود است، انرژی اضافه برای بازه‌های زمانی که کمبود دریافت انرژی وجود دارد ذخیره می‌گردد و در هر بازه‌ی زمانی امکان استفاده از کل انرژی موجود فراهم است. بنابراین، با لحاظ نمودن مفروضات صورت قضیه هیچگاه با کمبود انرژی و زمان مواجه نخواهیم شد و الگوریتم بهینه است.

### ۴-۳- EAS<sup>C</sup> برای مخزن با ظرفیت محدود

در شرایطی که ظرفیت مخزن محدود باشد امکان سرریز انرژی وجود خواهد داشت. این اتفاق در شرایطی رخ می‌دهد که مجموع انرژی موجود در مخزن و انرژی برداشت شده از محیط با احتساب انرژی مصرفی توسط پردازنده از ظرفیت مخزن بیشتر باشد. هدف اصلی الگوریتم  $EAS^C$  کمینه کردن میزان هدررفت انرژی است.

همان‌طور که پیش‌تر نیز ذکر شد، هدف اصلی  $EAS^\infty$  کمینه کردن انرژی مصرفی در حین مشاهده‌ی تمام موعدهای ممکن و ذخیره‌ی هر چه بیشتر انرژی برداشتی در مخزن انرژی در راستای اجرای کارهای آتی است. لیکن، مصرف کمینه‌ی انرژی برای سیستمی با ظرفیت محدود مخزن لزوماً تصمیم بهینه نخواهد بود. از طرف دیگر، در مواقعی که مخزن انرژی پر است، مصرف کمینه‌ی انرژی ممکن است منجر به هدر رفتن انرژی برداشتی از محیط شود. لذا، هدف تطابق الگوی مصرف انرژی با الگوی برداشت انرژی در راستای استفاده‌ی کارآمد از انرژی در دسترس است. برای این کار، ما به تخمین بازه‌های زمانی می‌پردازیم که انرژی قابل ذخیره‌سازی از ظرفیت مخزن بیشتر است. سپس، از انرژی که امکان هدر رفتن دارد در برش‌های زودتر و برای اجرای کارهای بیشتر استفاده می‌گردد. در واقع  $EAS^C$  مصالحه‌ای بین صرفه‌جویی در انرژی و زمان مصرفی انجام می‌دهد.

در ادامه با بکارگیری الگوریتم ارائه شده در شکل ۳، الگوریتم مربوط به مخزن با ظرفیت محدود را ارائه می‌کنیم. در شکل ۴، با شروع از برش جاری و در نظر گرفتن انرژی در دسترس و انرژی مورد نیاز هر برش، در یک فرآیند تکراری پیش می‌رویم تا برش  $slot_k$  که در آن مخزن پر می‌شود شناسایی گردد. سپس در خط ۸ انرژی اضافی بین برش‌های  $\{slot_{s1}, \dots, slot_k\}$  توزیع می‌گردد. با فراخوانی الگوریتم ارائه شده در شکل ۴ قبل از خط ۱۵ شکل ۳، انرژی اضافه‌ی مربوط به برش فعال جاری شناسایی می‌شود و فرکانس به‌دست آمده در خط ۱۲ شکل ۳ به‌هنگام می‌گردد.

با استفاده از این انرژی اضافه‌ی توزیع شده می‌توان کارهای بیشتری را در هر کدام از این برش‌ها اجرا نمود. بنابراین، کارها نسبت به الگوریتم شکل ۳ زودتر تمام می‌شوند. به این ترتیب برای کارهای بعدی زمان آزاد بیشتری خواهیم داشت و در واقع در اینجا به‌جای صرفه‌جویی در انرژی، در مصرف زمان صرفه‌جویی کرده‌ایم. تابع  $distributeExtraEnergy()$  سعی در توزیع هم‌وزن انرژی اضافه با ملاحظات زیر خواهد داشت:

۱. میزان انرژی تخصیص داده شده به برش  $slot_i$  حداکثر برابر با میزان انرژی مورد نیاز برای اجرای مجموع سیکل‌های باقی‌مانده از کارهای ترخیص شده تا شروع این برش است،

بزرگتر از ۱ باشد به معنای حضور سیستم در وضعیت اضافه‌بار بوده و لذا با یک تصمیم‌گیری حریصانه آن کار مطابق خط ۱۶ دور انداخته می‌شود. پس از انجام بررسی‌های مربوط به کمبود انرژی و زمان در خط ۱۵، در صورت وجود کمبود در هر کدام از آن‌ها با بازگشت به خط ۱۲ مجدداً فرکانس محاسبه خواهد شد.

#### Algorithm 1 = $EAS^\infty$

```

1.  $T = \{T_i: (C_i, P_i) | 1 \leq i \leq n\}$ 
2.  $J = \{J_{i,j}: (r_{i,j}, R_{ex_{i,j}}, R_{D_{i,j}}) | 1 \leq i \leq n, 1 \leq j \leq \frac{H}{P_i}\}$ 
3.  $epochsNumber = H/L$  //number of slots in a hyperperiod
4. for  $i = 0$  to  $epochsNumber - 1$ 
5.   for all  $J_{i,j} \in J$  // check all jobs in job set to determine job deadline violation or job completion
6.     if  $R_{D_{i,j}} = 0$  and  $R_{ex_{i,j}} > 0$  then
7.        $J_{i,j}$  violates its deadline
8.     elseif  $R_{D_{i,j}} > 0$  and  $R_{ex_{i,j}} = 0$  then
9.       remove  $J_{i,j}$  from  $J$ 
10.    endifor
11.    $f = POP(J)$  // calculate the best frequency for the current job set.
12.    $J_{ED} = \arg(\min_{J_{i,j} \in J}(R_{D_{i,j}}))$  // earliest deadline job
13.    $f_{ED} = \frac{R_{ex_{ED}}}{R_{D_{ED}}}$  //minimum required frequency for earliest deadline job to be completed before its deadline
14.   if  $Pow(f_{ED}) \times R_{D_{ED}} > e_b(i \times L) + e_h(i \times L, i \times L + R_{D_{ED}})$  or  $(f > 1)$ 
15.     then
16.       remove the earliest deadline job from job set and return to line 12
17.     if  $Pow(f) \times L > e_b(i \times L) + e_h(i \times L, (i + 1) \times L)$ 
18.       // If we don't have enough energy to work with frequency  $f$ , we need to calculate a new frequency corresponding to the available energy
19.       then
20.          $f =$  maximum frequency that can be obtained from  $e_b(i \times L) + e_h(i \times L, (i + 1) \times L)$  amount of energy
21.       endif
22.        $R_{ex_{ED}} -= f \times L$  //update the remaining execution cycle of earliest deadline job
23.       for  $J_{i,j} \in J$  // update the remaining relative deadline of released jobs
24.         if  $r_{i,j} \leq i \times L$  then  $R_{D_{i,j}} -= L$ 
25.       endfor
26.       Set the timer to expire at  $((i + 1) \times L)$  and sleep
27.   endfor

```

شکل ۳- الگوریتم  $EAS^\infty$

سپس، از خط ۱۷ تا ۱۹، انرژی در دسترس در برش  $slot_i$  مورد بررسی قرار می‌گیرد. مجموع مقادیر  $e_b(i \times L)$  و  $e_h(i \times L, (i + 1) \times L)$  بیانگر کل انرژی مورد انتظار در  $slot_i$  خواهد بود. اگر میزان انرژی در دسترس از انرژی مورد نیاز کمتر باشد در خط ۱۸ فرکانس جدید متناظر با آن محاسبه خواهد شد. در خط ۲۰ از زمان بند هم محاسبه مربوط به کار با زودترین موعد با فرکانس  $f$  انجام شده و مقدار باقی‌مانده‌ی سیکل‌های اجرای آن پس از برش جاری ثبت می‌گردند. در خطوط ۲۱ الی ۲۳ نیز تمام کارهای مجموعه‌ی [مورد بررسی قرار گرفته و موعد کلیه‌ی کارهای ترخیص شده برای لحظه‌ی انتهای برش جاری به‌هنگام می‌شوند. یک تایمر وظیفه‌ی فعال نمودن زمان‌بند در مرز برش‌ها را بر عهده خواهد داشت که در خط ۲۴ مقداردهی شده و پس از آن، الگوریتم زمان‌بندی تا ابتدای برش بعد غیرفعال می‌شود. این الگوریتم در هر ابردوره تکرار می‌گردد.

در ضمن، یک بازه ی زمانی با  $I_i = \{[s_i, e_i], f_i, j_i\}$  نمایش داده می شود که  $I = \{I_i\}$  نیز مجموعه ی تمام بازه های قابل تعریف است. فرکانس  $f_i$  برای یک بازه ی  $I_i$  حداقل فرکانس مورد نیاز پردازنده برای اتمام کارهای ترخیص شده درون آن بازه است که باید قبل از اتمام بازه به پایان برسند.  $I_i$  نیز زیرمجموعه ای از  $J$  است که زمان ترخیص آن ها بزرگتر یا مساوی  $s_i$  و موعد آن ها کوچکتر یا مساوی  $e_i$  است.  $|I_i| = e_i - s_i$  هم نیز طول بازه را نشان می دهد که بر این اساس خواهیم داشت:

$$f_i = \frac{\sum_{j \in I_i} R_{exj}}{|I_i|} \quad (5)$$

تعریف دیگری بنام بازه ی بحرانی نیز برای به دست آوردن زمان بندی با حداقل انرژی مصرفی برای یک مجموعه کار ارائه می شود. یک بازه ی بحرانی  $I_i$  بازه ای است که در هر زمان بندی بهینه برای مجموعه  $J$  باید گروهی از کارها در آن بازه در فرکانس حداکثر و ثابت  $f_i$  زمان بندی شوند. بنا به الگوریتم LPEDF، در ابتدا، یک بازه ی بحرانی  $I_i$  شناسایی می شود، سپس، کارهای بحرانی در فرکانس  $f_i$  در این بازه زمان بندی می شوند، بعد از آن، یک مساله کوچکتر برای کارهای باقی مانده تعریف می شود، و همین روند به صورت بازگشتی ادامه می یابد.

در ادامه اثبات می شود که برای پیدا کردن یک بازه ی بحرانی در یک زمان بندی بهینه برای وظایف شبه دوره ای، کافی است به دنبال بازه هایی بگردیم که زمان آغاز آن ها با زمان آغاز بازه ی زمان بندی یکسان باشد. جزئیات الگوریتم POP در شکل ۵ ارائه شده است.

**توضیح الگوریتم:** این الگوریتم یک مجموعه کار  $J$  را که باید در بازه ی زمانی  $[S, E]$  و  $S$  به ترتیب زمان های آغاز و پایان بازه ی زمان بندی هستند اجرا شوند به عنوان ورودی دریافت کرده و یک زمان بندی با قابلیت اجرای به هنگام تمام کارها و مصرف کمینه ی انرژی برمی گرداند.

#### Algorithm 3: Schedule = POP ( $J, [S, E]$ )

1. **Input:** the job set  $J$  and scheduling interval  $[S, E]$
2. **Output:** optimal schedule
3. **while** ( $J \neq \emptyset$ ) **do**
4.      $I =$  the set of all possible intervals in  $[S, E]$
5.      $I_s = \{I_i \in I \mid s_i = S\}$
6.      $I_c = \arg(\max_{I_i \in I_s} (f_i))$
7.     Remove all jobs  $J_c$  from  $J$
8.     Add  $I_c$  to the previous Schedule
9.     **for**  $p_i \in P$  where  $p_i > s_c$  **do**
10.         **if**  $p_i \leq e_c$  **then**  $p_i = s_c$
11.         **else**  $p_i = p_i - |I_c|$  **end if**
12.     **endfor**
13. **endwhile**
14. **return** Schedule

شکل ۵- الگوریتم POP

خط ۵، مجموعه ی  $I_s$  را معرفی می کند که اعضای آن بازه های با زمان آغاز برابر با زمان آغاز زمان بندی هستند. ایده ی اصلی، استفاده از یک روش تکراری برای شناسایی بازه های بحرانی موجود در  $I_s$  (خط ۶)، زمان بندی آن بازه های بحرانی (خطوط ۷ و ۸) و حذف آن ها از مجموعه ی کارها (خطوط ۹ تا ۱۲) است. در پایان نیز خط ۱۴ زمان بندی بهینه مورد استفاده در الگوریتم شکل ۳ را برمی گرداند.

**لم ۱ (اثبات بهینه بودن الگوریتم POP):** برای پیدا کردن بازه ی بحرانی از بین تمام بازه های ممکن کافی است بازه های با زمان آغاز  $S$  را جستجو کنیم. بنابراین

۲. حداکثر انرژی تخصیص داده شده به برش  $slot_i$  برابر با پیش بینی انجام شده برای انرژی موجود در این برش است.

#### Algorithm 2: EAS<sup>C</sup> ( $k$ )

1. **Input:**  $k =$  index of active slot
2.  $c = e_b(k \times L)$  // stored energy in the storage at the start time of  $slot_k$
3.  $si = k$  // start index  $k$  is saved in a temporary variable  $si$
4. **Do**
5.      $c = c + e_n(k \times L, (k + 1) \times L) - Pow(f) \times L$   
      // Check  $slot_k$  available energy in the storage
6.      $k = k + 1$
7. **While** ( $c \leq C$ ) //continue until storage is not full
8. **distributeExtraEnergy** ( $c - C, \{slot_{si}, \dots, slot_k\}$ )

شکل ۴- الگوریتم EAS<sup>C</sup>

## ۵- یک الگوریتم زمان بندی با مصرف بهینه انرژی برای وظایف شبه دوره ای

در این بخش، پس از معرفی رده ای از وظایف به نام وظایف شبه دوره ای، یک الگوریتم زمان بندی برای مصرف بهینه ی انرژی توسط این وظایف ارائه می شود. دو خصوصیت مهم برای وظایف شبه دوره ای متصور است: یکی اینکه لحظه ترخیص کار اول یک وظیفه لزوماً لحظه ی صفر نیست، و دوم اینکه کار اول یک وظیفه دارای بهره وری (حاصل تقسیم زمان اجرای باقی مانده به موعد نسبی باقی مانده) بزرگتری نسبت به سایر کارهای آن وظیفه است.

دلیل تعریف وظایف شبه دوره ای این است که مدل وظایف سیستم مورد بحث در این مقاله دوره ای است، لیکن ممکن است با گذشت زمان به مدل شبه دوره ای تغییر ماهیت دهند. در ابتدای هر ابر دوره (برابر با ابتدای  $slot_0$ ) تمام وظایف دوره ای هستند. لذا، می توان با استفاده از الگوریتم زمان بندی DVFS کلاسیک برای وظایف دوره ای T [6]، فرکانس بهینه را به صورت زیر محاسبه نمود:

$$f = \sum_{T_k \in T} C_k / P_k \quad (3)$$

در صورتیکه وظایف قابل زمان بندی باشند، به سادگی می توان یک فرکانس بهینه برای کل ابر دوره محاسبه نمود تا انرژی مصرفی کمینه گردد. لیکن ممکن است به دلیل غیرقطعی بودن پیش بینی میزان انرژی برداشتی و کمبود انرژی در یک برش  $slot_i$  امکان بکارگیری فرکانس ابتدایی تعیین شده در آن برش وجود نداشته باشد. در این شرایط باید تمام انرژی موجود در آن برش را با فرکانسی پایین تر از فرکانس بهینه مصرف نمود. بنابراین، از آنجا که موفق به اجرای میزان کار مورد انتظار در برش  $slot_i$  نشده ایم، ادامه کار در برش  $slot_{i+1}$  با فرکانس بهینه ی برش قبل منجر به رعایت مواعدها نخواهد شد و باید فرکانس بهینه ی برش جدید را مجدداً محاسبه نماییم. به این ترتیب وظایف از حالت دوره ای به شبه دوره ای تبدیل می شوند.

در ادامه، با استفاده از ایده ی الگوریتم LPEDF [۱۵]، یک الگوریتم زمان بندی برای وظایف شبه دوره ای ارائه می کنیم. در ابتدا برخی نمادگذاری های مورد نیاز را معرفی می نماییم. مجموعه ی  $P$  شامل زمان های ترخیص و موعد برای  $n$  کار مجموعه ی  $J$  در کنار زمان های آغاز  $(S)$  و اتمام  $(E)$  بازه ی زمان بندی است که به این ترتیب حداکثر  $2n + 2$  عنصر خواهد داشت:

$$P = \{S, r_1, r_1 + R_{D1}, r_2, r_2 + R_{D2}, \dots, r_n, r_n + R_{Dn}, E\} \quad (4)$$

جدول ۲- فرکانسها و توان مصرفی پردازنده‌ی Intel Xscale [21]

Frequency (MHz)	Power Consumption (mW)
150	80
400	400
600	1000
800	2000
1000	3200

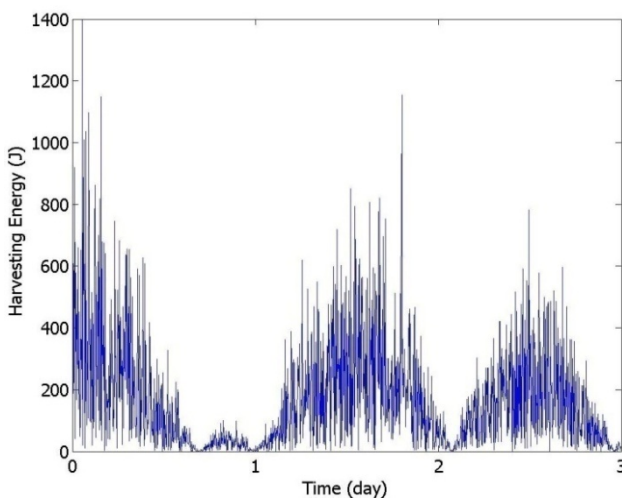
تعداد ۱۰۰ مجموعه وظیفه با بهره‌وری‌های مختلف پردازنده از ۰.۱ تا ۱.۰ با گام‌های ۰.۱ در نظر گرفته شده‌اند. هر مجموعه از وظایف بین ۵ تا ۱۵ وظیفه دارد که دوره‌ی آن‌ها بصورت تصادفی از مجموعه‌ی  $\{10, 20, \dots, 100\}$  انتخاب شده است. بیشترین سیکل مورد نیاز برای هر وظیفه نیز بصورت تصادفی و بر اساس بهره‌وری پردازنده انتخاب می‌شود. برای انرژی برداشتی از محیط  $(e_h(t))$  نیز یک الگوی تصادفی بر اساس [۱۰] و به شرح زیر مورد استفاده قرار گرفته است:

$$e_h(t) = \left[ 100 \times N(t) \times \cos\left(\frac{t}{70\pi}\right) \times \cos\left(\frac{t}{100\pi}\right) \right] \quad (9)$$

که  $N(t)$  یک متغیر تصادفی با توزیع نرمال با میانگین صفر و واریانس ۱ است. شکل ۶ رفتار تواما تصادفی و دوره‌ی انرژی برداشتی از محیط را نشان می‌دهد که منظور از رفتار دوره‌ی شبیه‌سازی انرژی خورشید در شبانه‌روز است. در انجام آزمایشات، الگوریتم‌های  $EAS^\infty$  و  $EAS^C$  با سه الگوریتم زمان‌بندی زیر مورد مقایسه قرار گرفته‌اند:

۱- EDF که کارها را با حداکثر فرکانس اجرا می‌کند و در زمان‌های نبود انرژی یا کار آماده، پردازنده را به حالت آماده‌به‌کار می‌برد،  
 ۲- staticEDF [۶] که یک الگوریتم زمان‌بندی مبتنی بر DVFS برای وظایف دوره‌ی است،

۳- EA-DVFS [۷] که یک الگوریتم انرژی-آگاه مبتنی بر DVFS برای سیستم‌های برداشت گر انرژی است. این الگوریتم در صورت وجود انرژی ذخیره‌شده‌ی کافی، پردازنده را با حداکثر سرعت به کار می‌اندازد که بر اساس انرژی موجود در سیستم برای هر کار یک زمان شروع محاسبه می‌کند.



شکل ۶- الگوی تصادفی انرژی برداشتی  $e_h(t)$

شکل ۷ بر اساس میزان انرژی برداشتی در طول شبانه‌روز  $(\overline{e_h(t)})$  درصد کارهایی را نشان می‌دهد که موعد خود را از دست می‌دهند. در این آزمایش برای الگوریتم با مخزن محدود،  $C = 100$  kJ و بهره‌وری پردازنده برابر با ۰.۷ در نظر

الگوریتم POP بازه‌ی بحرانی و فرکانس بهینه را برای برش آتی مجموعه کار [ ] با خاصیت  $n = ||\theta||$  با پیچیدگی  $\theta(n)$  پیدا می‌کند.

اثبات: در ابتدای ابردوره وظایف دوره‌ی هستند و فرکانس بهینه برابر است با:

$$f_0 = \sum_{T_k \in T} C_k / P_k \quad (6)$$

بنابراین، زمان‌بندی بهینه برای کل ابردوره عبارتست از:

$$S = \{([S, E], f_0, J)\} \quad (7)$$

فرض کنید زمان‌بند در لحظه‌ی صفر کار خود را آغاز کرده باشد و در لحظه‌ی  $t = i \times L$  مجموعه‌ی تمام بازه‌های ممکن  $R_t = \{[t_i, ( [s_{t,i}, e_{t,i}], f_{t,i}, J_{t,i}) | 0 \leq i \leq m_t\}$  با در نظر گرفتن  $R_0 = \{[0, ( [s_{0,i}, e_{0,i}], f_{0,i}, J_{0,i}) | 0 \leq i \leq m_0\}$  به عنوان مجموعه‌ی تمام بازه‌های ممکن در بازه‌ی  $[0, H]$ ، با استفاده از (7) می‌توان به این جمع بندی رسید که  $[0, H]$  یک بازه‌ی بحرانی است و بیشترین فرکانس را داراست. بنابراین، نامساوی زیر برقرار خواهد بود:

$$f_{0,1}, f_{0,2}, \dots, f_{0,m_0} \leq f_0 \quad (8)$$

اگر بخواهیم از فرکانس  $f_0$  در برش  $slot_0$  استفاده کنیم، دو حالت پیش می‌آید:

$$1. \text{Pow}(f_0) \times L \leq e_b(t) + e_h(t, t+L)$$

$$2. \text{Pow}(f_0) \times L > e_b(t) + e_h(t, t+L)$$

در حالت اول با فرکانس  $f_0$  کار می‌کنیم و بنابراین نیازی به محاسبه‌ی فرکانس بهینه در برش  $slot_1$  نخواهد بود. لیکن، در حالت دوم باید بازه‌ی بحرانی و فرکانس بهینه را مجدداً محاسبه نمود. فرض کنیم  $R_1 = \{[1, ( [s_{1,i}, e_{1,i}], f_{1,i}, J_{1,i}) | 0 \leq i \leq m_1\}$  ممکن در برش  $slot_1$  باشد. در ضمن، مجموعه‌ی بازه‌های مشترک  $R_0$  و  $R_1$  را با  $R_c$  نمایش می‌دهیم. بازه‌هایی که در  $R_1$  موجود هستند ولی در  $R_c$  نیستند، در آغاز برش  $slot_1$  شروع می‌شوند. از آنجا که در برش  $slot_0$  سیستم با فرکانس  $f_0$  کار نکرده است، واضح است که این بازه‌ها فرکانسی بیش از  $f_0$  خواهند داشت. از آنجا که تمام بازه‌های  $R_c$  فرکانسی کمتر از  $f_0$  دارند، هیچ بازه‌ی بحرانی در  $R_c$  پیدا نخواهد شد. بنابراین کافی است به جستجو در  $R_1 - R_c$  بپردازیم. به عبارت دیگر کافی است فقط بازه‌هایی بررسی شوند که زمان آغاز آن‌ها همان زمان آغاز کل بازه‌ی زمان‌بندی است. همین اثبات برای تمام برش‌های  $slot_i$  قابل بیان خواهد بود و به این ترتیب محاسبه پیچیدگی زمانی  $\theta(n)$  بسادگی انجام می‌گردد.

## ۶- نتایج شبیه‌سازی

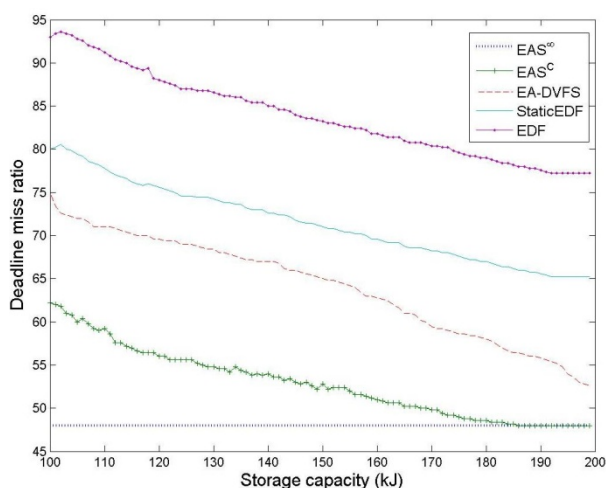
در این بخش به ارائه‌ی برخی آزمایش‌ها برای تعیین میزان کارآمدی الگوریتم‌های پیشنهادی می‌پردازیم. برای این منظور یک پردازنده‌ی مجهز به DVFS از خانواده‌ی Intel XScale با پنج سطح سرعت در نظر می‌گیریم. مشخصات مورد نیاز از این پردازنده در جدول ۲ نشان داده شده‌اند.

گرفته شده است. در ضمن، مقدار اولیه‌ی انرژی ذخیره شده در باتری نیز  $e_b(0) = 100\text{kJ}$  است. همانطور که در شکل ۷ دیده می‌شود، هر دو الگوریتم  $EAS^\infty$  و  $EAS^C$  بهتر از الگوریتم‌های EDF، staticEDF و EA-DVFS عمل می‌کنند. از آنجا که الگوریتم‌های EDF و staticEDF برای سیستم‌های برداشت‌گر انرژی با خاصیت تصادفی انرژی برداشتی طراحی نشده‌اند، بیش از دیگر الگوریتم‌ها موعد از دست می‌دهند. در ضمن، دلایل عملکرد ضعیف‌تر EA-DVFS نسبت به الگوریتم‌های پیشنهادی در این مقاله به شرح زیر است:

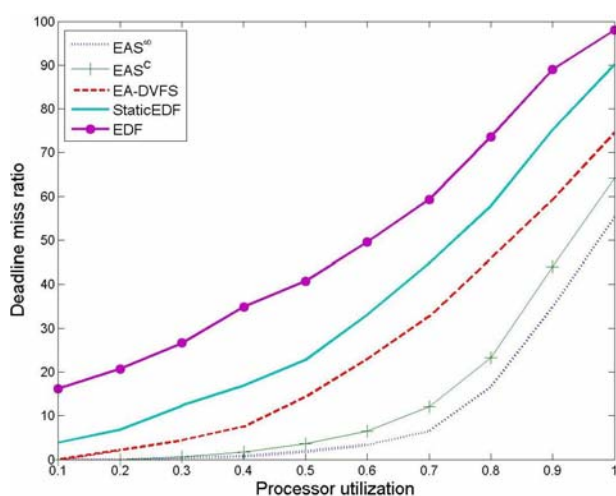
- برای EA-DVFS کارهای با زمان ترخیص دلخواه طراحی شده است و قادر نیست بهینه‌سازی‌های الگوریتم‌های پیشنهادی این مقاله برای وظایف دوره‌ای را انجام دهد.
- الگوریتم‌های زمان بندی مبتنی بر DVFS به طور مرسوم از باقی مانده‌ی زمان اجرا و موعد تمام کارهای ترخیص شده تا لحظه‌ی تصمیم‌گیری برای انتخاب فرکانس استفاده می‌کنند. الگوریتم EA-DVFS فقط فرکانس حداقل برای کار با زودترین موعد را محاسبه می‌کند به طوریکه ممکن است تمام انرژی موجود تا موعد همین کار تخلیه گردد. لیکن، مصرف کمینه‌ی انرژی برای پراولویت‌ترین کار ممکن است به کم شدن زمان در دسترس برای سایر کارها و بنابراین از دست رفتن موعد آنها منجر شود.

این در حالی است که الگوریتم‌های پیشنهادی ما از تمام اطلاعات کارهای ابردوره برای پیدا کردن بهترین فرکانس استفاده می‌کنند. به عبارت دیگر، همان طور که در شکل ۷ قابل مشاهده است، در شرایط زیاد بودن انرژی برداشتی کارایی این الگوریتم‌ها تفاوت چندانی با هم ندارند. لیکن، در شرایطی که انرژی برداشتی در طول شبانه‌روز چندان زیاد نیست، میزان بهبود الگوریتم  $EAS^C$  نیز در کنار  $EAS^\infty$  قابل ملاحظه است.

دلیل این امر بدین شرح است که  $EAS^C$  در شرایط سرریز شدن مخزن با افزایش سرعت (عدم صرفه‌جویی در انرژی)، در زمان صرفه‌جویی ایجاد می‌کند: بدین ترتیب درحالی‌که از انرژی برداشتی بهتر بهره‌برداری می‌نماید، کارها را نیز در زمان کوتاه‌تری خاتمه می‌دهد و با ایجاد لختی بیشتر، امکان بهره‌گیری از DVFS برای اجرای سایر کارها با سرعت کمتر و مشاهده موعد آنها را در شرایطی که در برش‌های آبی انرژی برداشتی کمتر از پیش‌بینی باشد فراهم می‌کند. حال آنکه الگوریتم‌هایی مثل EA-DVFS از این قابلیت برخوردار نیستند و صرفاً به صرفه‌جویی انرژی می‌پردازند و در شرایط سرریز مخزن، انرژی برداشتی را هدر می‌دهند.



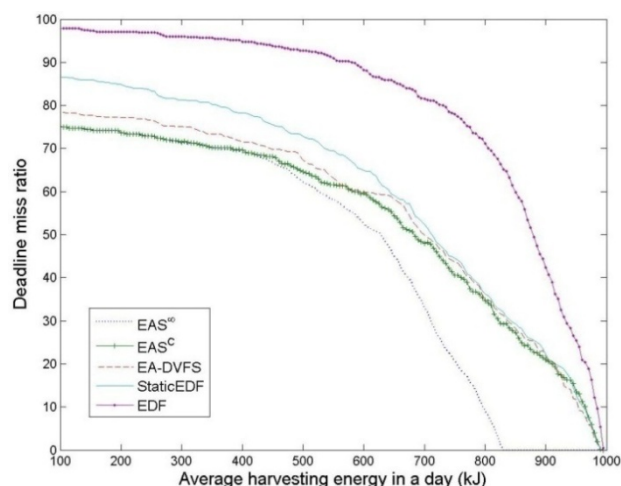
شکل ۸- مقایسه‌ی درصد موعدهای از دست رفته در ظرفیت‌های مختلف مخزن



شکل ۹- مقایسه‌ی درصد موعدهای از دست رفته در بهره‌وری‌های مختلف پردازنده

این در حالی است که الگوریتم‌های پیشنهادی ما از تمام اطلاعات کارهای ابردوره برای پیدا کردن بهترین فرکانس استفاده می‌کنند. به عبارت دیگر، همان طور که در شکل ۷ قابل مشاهده است، در شرایط زیاد بودن انرژی برداشتی کارایی این الگوریتم‌ها تفاوت چندانی با هم ندارند. لیکن، در شرایطی که انرژی برداشتی در طول شبانه‌روز چندان زیاد نیست، میزان بهبود الگوریتم  $EAS^C$  نیز در کنار  $EAS^\infty$  قابل ملاحظه است.

دلیل این امر بدین شرح است که  $EAS^C$  در شرایط سرریز شدن مخزن با افزایش سرعت (عدم صرفه‌جویی در انرژی)، در زمان صرفه‌جویی ایجاد می‌کند: بدین ترتیب درحالی‌که از انرژی برداشتی بهتر بهره‌برداری می‌نماید، کارها را نیز در زمان کوتاه‌تری خاتمه می‌دهد و با ایجاد لختی بیشتر، امکان بهره‌گیری از DVFS برای اجرای سایر کارها با سرعت کمتر و مشاهده موعد آنها را در شرایطی که در برش‌های آبی انرژی برداشتی کمتر از پیش‌بینی باشد فراهم می‌کند. حال آنکه الگوریتم‌هایی مثل EA-DVFS از این قابلیت برخوردار نیستند و صرفاً به صرفه‌جویی انرژی می‌پردازند و در شرایط سرریز مخزن، انرژی برداشتی را هدر می‌دهند.



شکل ۷- مقایسه‌ی درصد موعدهای از دست رفته در سطوح مختلف انرژی برداشتی



*International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, pp. 125-134, 2008.

[10] C. Moser, D. Brunelli, L. Thiele, and L. Benini, "Real-time scheduling for energy harvesting sensor nodes," *Real-Time Systems*, vol. 37, pp. 233-260, 2007.

[11] A. Kansal, D. Potter, and M. B. Srivastava, "Performance aware tasking for environmentally powered sensor networks," *Proceedings of the joint international conference on Measurement and modeling of computer systems*, pp. 223-234, 2004.

[12] C. Bergonzini, D. Brunelli, and L. Benini, "algorithms for harvested energy prediction in batteryless wireless sensor networks," *IWASI 2009. 3rd International Workshop on Advances in sensors and Interfaces*, pp.144-149, 2009.

[13] C. Moser, L. Thiele, D. Brunelli, and L. Benini, "Adaptive power management in energy harvesting systems," *Proceedings of the conference on Design, automation and test in Europe*, pp. 773-778, 2007.

[14] A. Kansal, J. Hsu, S. Zahedi, and M.B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 6, pp. 32-66, 2007.

[15] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced CPU energy," *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, pp. 374-382, 1995.

[16] X. Zhong, and C. Z. Xu, "Energy-aware modeling and scheduling for dynamic voltage scaling with statistical real-time guarantee," *IEEE Transactions on computers*, vol. 56, no. 3, pp. 358-372, 2007.

[17] C. Rusu, R. Melhem, and D. Mossé, "Multi-version scheduling in rechargeable energy-aware real-time systems," *Journal of Embedded Computing*, vol. 1, pp. 271-283, 2005.

[18] A. Allavena, and D. Mossé, "Scheduling of frame-based embedded systems with rechargeable batteries," *Workshop on power management for real-time and embedded systems (in conjunction with RTAS 2001)*, 2001.

[19] S. Liu, J. Lu, Q. Wu, and Q. Qiu, "load-matching adaptive task scheduling for energy efficiency in energy harvesting real-time embedded systems," *ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED)*, pp. 325-330, 2010.

[20] S. Liu, J. Lu, Q. Wu, and Q. Qiu, "Harvesting-aware power management for real-time systems with renewable energy," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, pp. 1473-1486, 2012.

[21] D. Zhu, N. AbouGhazaleh, D. Mosse, and R. Melhem, "power aware scheduling for and/or graphs in multi-processor real-time system," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 9, pp. 849-864, 2004.

## ۷- نتیجه گیری

در این مقاله دو الگوریتم زمان بندی برای سیستم های بی درنگ برداشت گر انرژی از محیط ارائه گردید: یکی برای مخزن با ظرفیت نامحدود و دیگری برای مخزن با ظرفیت محدود. در شرایط دارا بودن امکان پیش بینی دقیق انرژی برداشتی، بهینه بودن الگوریتم اول اثبات گردید. در ضمن نشان داده شد در شرایطی که امکان تامین مخزن بزرگ انرژی وجود نداشته باشد، الگوریتم دوم قادر است با جلوگیری از سرریز انرژی در میزان زمان مصرفی صرفه جویی کند. نتایج شبیه سازی ها حاکی از برتری الگوریتم های پیشنهادی نسبت به EDF، staticEDF و EA-DVFS برای مقادیر مختلف ظرفیت مخزن انرژی، بهره وری پردازنده، و انرژی برداشتی است.

## مراجع

[1] J. A. Paradiso, and T. Starner, "Energy scavenging for mobile and wireless electronics," *IEEE Pervasive computing*, vol. 4, no. 1, pp. 18-27, 2005.

[2] K. Lin, J. Yu, J. Hsu, S. Zahedi, D. Lee, J. Friedman, A. Kansal, V. Raghunathan, and M. Srivastava, "Heliomote: enabling long-lived sensor networks through solar energy harvesting," *Proceedings of the 3rd international conference on Embedded networked sensor systems*, pp. 309-317, 2005.

[3] H. Dubois-Ferrière, L. Fabre, R. Meier, and P. Metrailler, "TinyNode: a comprehensive platform for wireless sensor network applications," *Proceedings of the 5th international conference on Information processing in sensor networks*, pp. 358-365, 2006.

[4] C. Moser, L. Thiele, D. Brunelli, and L. Benini, "adaptive power management for environmentally powered systems," *IEEE Transactions on Computers*, vol. 59, no.4, pp.478-491, 2010.

[5] S. Roundy, D. Steingart, L. Frechette, P. Wright, and J. Rabaey, "Power sources for wireless sensor networks," *Wireless Sensor Networks*, pp. 1-17, 2004.

[6] P. Pillai, and K. G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," *Proceedings of the eighteenth ACM symposium on Operating systems principles*, vol. 35, no. 5, pp. 89-102, 2001.

[7] S. Liu, Q. Qiu, and Q. Wu, "Energy aware dynamic voltage and frequency selection for real-time systems with energy harvesting," *Proceedings of the conference on Design, automation and test in Europe*, pp. 236-241, 2008.

[8] J. Recas, C. Bergonzini, D. Atienza, and T. S. Rosing, "Prediction and management in energy harvested wireless sensor nodes," *Proceedings of the 1st international conference on Wireless communication*, pp. 6-10, 2009.

[9] A. E. Susu, A. Acquaviva, D. Atienza, and G. De Micheli, "Stochastic modeling and analysis for environmentally powered wireless sensor nodes," *6th*

## ضمیمه ی الف

## واژه نامه

Feasible Schedule	زمان بندی ممکن	Hyperperiod	ابردوره
Slack	لختی	Overload	اضافه بار
Sporadic	پراکنده	Harvesting Energy-Aware	آگاه از انرژی برداشتی
Deadline	موعد	Hard Real-Time	بی درنگ سخت
Harvesting Energy-Unaware	ناآگاه از انرژی برداشتی	Firm Real-Time	بی درنگ محکم
Deadline Miss	نقض موعد	Soft Real-Time	بی درنگ نرم
Embedded	نهفته	Harvesting	برداشت



**مهدی کارگهی** از سال ۱۳۸۶ عضو هیئت علمی دانشکده مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران و سرپرست آزمایشگاه تحقیقاتی سیستم های بی درنگ و اتکاپذیر دانشگاه تهران است. او از سال ۱۳۸۱ با پژوهشکده علوم کامپیوتر پژوهشگاه علوم بنیادی (IPM) نیز همکاری تحقیقاتی داشته است. علائق تحقیقاتی وی مدل سازی و مدیریت کارایی، انرژی و دما در سیستم های بی درنگ، توزیع شده، و اتکاپذیر است.

آدرس پست الکترونیکی ایشان عبارت است از:

kargahi@ut.ac.ir, kargahi@ipm.ir



**مریم دهقان منشادی** فارغ التحصیل مقاطع کارشناسی (سال ۱۳۸۷) و کارشناسی ارشد (سال ۱۳۸۹) از دانشکده مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران بوده است. وی دوره کارشناسی ارشد خود را در آزمایشگاه تحقیقاتی سیستم های بی درنگ و اتکاپذیر دانشگاه تهران و با تمرکز روی زمینه های تحقیقاتی مرتبط با سیستم های بی درنگ نهفته و برداشت گره های انرژی سپری کرده است.

آدرس پست الکترونیکی ایشان عبارت است از:

m.dehghan@ece.ut.ac.ir

## اطلاعات بررسی مقاله:

تاریخ ارسال: ۹۱/۴/۱۲

تاریخ اصلاح: ۹۱/۹/۱۳

تاریخ قبول شدن: ۹۱/۱۰/۱۷

نویسنده مرتبط: دکتر مهدی کارگهی، دانشکده مهندسی برق و کامپیوتر، پردیس دانشکده های فنی، دانشگاه تهران، تهران، ایران.

<sup>1</sup>Pseudo-Periodic Optimal Power-Aware Scheduling Algorithm