

# SID



سرویس های ویژه



سرویس ترجمه تخصصی



کارگاه های آموزشی



بلاگ مرکز اطلاعات علمی



سامانه ویراستاری STES



فیلم های آموزشی

## کارگاه های آموزشی مرکز اطلاعات علمی جهاد دانشگاهی

کارگاه آنلاین  
بررسی مقابله ای متون (مقدماتی)

کارگاه آنلاین  
پروپوزال نویسی و پایان نامه نویسی

کارگاه آنلاین آشنایی با پایگاه های اطلاعات علمی بین المللی و ترند های جستجو

# روشی مبتنی بر تطبیق الگو برای تخمین بیشترین زمان اجرای حلقه ها

مهدی سخائی نیا<sup>۱</sup>، سعید پارسا<sup>۲</sup>

<sup>۱</sup> گروه کامپیوتر، دانشکده مهندسی، دانشگاه بوعلی سینا، همدان، sakhaei@basu.ac.ir

<sup>۲</sup> گروه نرم افزار دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، persa@iust.ac.ir

## چکیده

برای یافتن یک زمانبندی ممکن در سیستم های بی درنگ باید بیشترین زمان اجرای هر وظیفه مشخص باشد تا تضمین گردد اجرای وظیفه قبل از پایان زمان اختصاص یافته به آن خاتمه می پذیرد. اجرای حلقه های تکرار بیشترین مدت زمان اجرای یک وظیفه یا برنامه را به خود اختصاص می دهند. این امر اهمیت تخمین بیشترین زمان اجرای حلقه ها را نشان می دهد. روش تطبیق الگو یکی از روشهایی است که برای تخمین بیشترین زمان اجرای حلقه ها ارائه شده است. در این روش در صورتیکه حلقه با الگوی ارائه شده تطبیق داشت با استفاده از یک معادله تعداد تکرار محاسبه می گردد. در حقیقت از تولید مقدار شمارنده برای تمامی تکرارها جلوگیری می گردد. نقص روش تطبیق الگو وابستگی زیاد آن به الگو می باشد. این وابستگی به ساختار و محل شرط تست کننده شمارنده، محل، نحوه و تعدد تغییر شمارنده مرتبط می باشد. در این مقاله با مدلسازی جریان اطلاعات حلقه در قالب دو دسته عبارت نمادین نشاندهنده شرط تکرار و نحوه تغییر شمارنده از وابستگی به الگو کاسته شده است. با یک مثال توانایی روش ارائه شده نشان داده شده است.

## واژه های کلیدی

تخمین بیشترین زمان اجرا، تحلیل زمانی، حلقه های تکرار، سیستم های نهفته بی درنگ، تحلیل ایستای برنامه.

## ۱- مقدمه

در سیستم های بی درنگ، برای یافتن یک زمانبندی ممکن و اطمینان از خاتمه یک وظیفه در مهلت<sup>۱</sup> تعیین شده، زمانبند باید مدت زمان اجرای هر وظیفه را بداند. مدت زمان اجرای یک وظیفه در اجراهای مختلف آن بر روی یک بستر مشخص متفاوت می باشد. این تفاوت ناشی از ورودی های وظیفه و حالت سخت افزار می باشد. بنابراین برای یافتن یک زمانبندی درست و مطمئن، بیشترین زمان اجرای<sup>۲</sup> هر وظیفه باید تخمین زده شود [1].

بیشتر نرم افزارهای سیستم های نهفته بی درنگ مشتمل بر حلقه های تکرار زیادی می باشند. از همین روست که تخمین مطمئن و نزدیک به مقدار واقعی بیشترین زمان اجرا برای حلقه های تکرار از اهمیت ویژه ای برخوردار است [2].

یکی از روشهای تخمین بیشترین زمان اجرای حلقه های تکرار روش تطبیق الگو<sup>۳</sup> می باشد [1]. روش تطبیق الگو بر پایه این واقعیت است که برای بیشتر حلقه های تکرار، کامپایلرها از مجموعه دستورات ماشین یکسان یا مشابه برای مقداردهی، روزرسانی و تست شمارنده حلقه استفاده می نمایند. روشهای تطبیق الگو اینگونه دستورات را در کد برنامه یافته و مقادیر عملوندهای این دستورات را تحلیل نموده تا بازه مقادیری که شمارنده می پذیرد را محاسبه نمایند. مقادیری که می توان به شمارنده

انتساب داد معمولا در قالب مقدار اولیه، میزان افزایش یا کاهش و مقدار نهایی شمارنده بیان می گردد. سپس می توان با استفاده از یک معادله بلافاصله تعداد تکرار حلقه را محاسبه نمود. نقص روشهای تطبیق الگو این است که اگر بهینه سازی کامپایلر انجام گردد یا کامپایلر خودش تغییراتی را داشته باشد، تطبیق الگو شکست خورده و نمی تواند تعداد تکرار را بدرستی محاسبه نماید. بطور کلی عدم تطبیق با الگو سبب شکست این روشهای می گردد [1]. سایر روشهای محاسبه تکرار که مبتنی بر تطبیق الگو نیستند باید مقدار شمارنده را در هر تکرار محاسبه نمایند و سپس تعداد تکرار را بدست آورند که زمان اجرای ابزار برای تحلیل بزرگ خواهد بود و کارایی کمی خواهد داشت [3,4].

الگو در روشهای تطبیق الگو وابسته به دو مجموعه دستورات خواهد بود. اولین مجموعه مربوط به دستوراتی است که شمارنده را مقداردهی و بروز رسانی می کنند یا اصطلاحاً چگونگی تغییر مقدار شمارنده را نشان می دهند. دومین مجموعه دستورات مربوط به تست شمارنده می باشد. این دستورات در غالب یک عبارت شرطی می باشد که می توان گفت شرط تکرار و یا خاتمه یک حلقه می باشند. این روشها اغلب محدود به موارد زیر هستند:

- ساختار و محل شرط تکرار و خاتمه
- محل و چگونگی تغییر شمارنده در بدنه حلقه تکرار
- تعداد متغیرهای شمارنده

که ترکیب عطفی عبارات شرطی بر روی این مسیر به مقدار درست ارزیابی شود. شرط تکرار از مسیر تکرار ip به صورت زیر تعریف می گردد:

$$ipCondition(ip, v, w) = \bigwedge_{bc \in ip} bc(v, w)$$

برای حذف وابستگی عبارت شرطی به نقاط بکار گرفته شده در برنامه، متغیرهای  $v_s$ ، با عبارات محاسباتی جایگزین می گردند که مقدار این متغیرها در  $w$  را بر اساس مقدار متغیرهایی،  $v_s$ ، در ابتدای حلقه تکرار محاسبه می کند. بنابراین شرط مسیر تکرار بدین صورت خواهد بود:

$$ipCondition(ip, v_s) = \bigwedge_{bc \in ip} bc(v_s)$$

برای هر مسیر پایان tp در یک حلقه نیز می تواند شرط مسیر را تعریف کرد:

$$tpCondition(tp, v_s) = \bigwedge_{bc \in tp} bc(v_s)$$

این عبارات شرطی را می توان در قالب یک دستگاه نامعادلات خطی نیز نشان داد.

درست یا نادرست بودن این شرطها وابسته به مقدار متغیرهای ظاهر شده در عبارات شرطی آن می باشد. تعدادی از این متغیرها در بدنه حلقه تکرار تغییر می نمایند که به آن شمارنده گفته می شود. در این مقاله این نوع متغیرها متغیرهای کنترلی حلقه نامیده شده است. انتخاب هر یک از این مسیرها به مقدار متغیرهای کنترلی حلقه در ابتدای حلقه تکرار وابسته می باشد که مقدار آنها از تکرار قبلی بدست آمده است. همچنین در تکرار بعدی مسیر تکرار بر اساس مقدار متغیرهای کنترلی در پایان این تکرار تعیین می گردد. بنابراین باید مشخص گردد که در هر مسیر تکرار نحوه تغییر متغیرهای کنترلی حلقه به چه صورت می باشد. عبارت نمادین نشان دهنده نحوه تغییر متغیرهای کنترلی حلقه تابعی خواهد بود که مقدار متغیرهای کنترلی را در انتهای مسیر تکرار فعلی  $cv_e$  (یا ابتدای مسیر تکرار بعدی) بر اساس مقدار متغیرهای کنترلی در ابتدای مسیر تکرار  $cv_s$  محاسبه می کند.

$$cv_e = chgVal(ip, cv_s)$$

پس برای هر مسیر تکرار ip در بدنه حلقه تکرار یک شرط تکرار ipCondition و یک عبارت نشان دهنده متغیرهای کنترلی chgVal خواهیم داشت. اجرای یک حلقه تکرار، ترکیبی (با تکرار) از این مسیرهای تکرار خواهد بود که به یکی از مسیرهای پایان ختم می گردد. برای جزئیات استخراج و ایجاد عبارت نمادین به [5] مراجعه گردد.

### ۳- تخمین بیشترین زمان اجرای حلقه

یک حلقه در هر تکرار از یک مسیر تکرار اجرا شده و در نهایت با اجرای روی یک مسیر پایان خاتمه می یابد. بنابراین زمان اجرای آن عبارت خواهد بود از:

$$T_{loop} = T_s + \sum_{i \in ip} T_i * X_i + T_{tp}$$

که  $T_s$  زمان دستوراتی است که قبل از اولین تکرار اجرا می گردند مانند مقداردهی اولیه شمارنده و  $T_i$  زمان اجرای دنباله دستورات روی

در این مقاله روشی نوینی ارائه شده است که تأثیری پذیری کمتری از محدودیت های فوق داشته باشد. برای حل این مشکل در این مقاله اطلاعات جریان کنترلی و داده ای حلقه تکرار استخراج گردیده و در قالب دو دسته عبارت نمادین مدل شده است. بدین منظور مسیرهای تکرار در بدنه حلقه که در هر تکرار، حلقه از یکی از آنها اجرا می شود شناسایی می گردد. برای هر مسیر تکرار یک عبارت شرط تکرار و یک عبارت چگونگی تغییر متغیرهای حلقه (شمارنده) ایجاد می گردد. عبارت شرط مسیر تکرار نشان می دهد با چه مقداری از متغیرها حلقه از این مسیر اجرا خواهد شد و عبارت چگونگی تغییر متغیرهای حلقه نشان می دهد که در این مسیر به مقدار هر یک از متغیرهای حلقه چه مقداری اضافه یا کم می گردد. سپس بر اساس این عبارات تعداد تکرار هر مسیر تکرار را بدست آورده و نهایتاً بیشترین زمان اجرای حلقه تکرار را محاسبه نمود.

در ادامه مقاله، در بخش دوم به نحوه مدلسازی جریان اطلاعاتی حلقه در قالب عبارت نمادین پرداخته شده است. در بخش سوم چگونگی محاسبه بیشترین زمان اجرا ارائه گردیده است. تخمین تعداد تکرار هر مسیر تکرار در بخش چهارم شرح داده شده است. بخش پنجم به بیان کارهای مرتبط پرداخته است. در بخش ششم مثالی ارائه شده است. توانایی روش ارائه شده در بخش هفتم ارزیابی گردیده است و در نهایت در بخش هشتم نتیجه گیری و کارهای آینده شرح داده شده است.

## ۲- مدلسازی اطلاعات جریانی حلقه

تعداد تکرار حلقه های مبتنی بر شمارنده وابسته به مقداری است که در طی تکرار حلقه به شمارنده انتساب می گردد. معمولاً در اینگونه حلقه ها، شمارنده با یک مقدار اولیه مقداردهی شده و در بدنه حلقه تکرار شمارنده اضافه یا کم می گردد. پس از اینکه شرطی مبتنی بر مقدار این شمارنده درست گردد، حلقه خاتمه می یابد. بنابراین چگونگی تغییر شمارنده در بدنه حلقه تکرار بر روی تعداد تکرارها تأثیر خواهد داشت. میزان کاهش/افزایش شمارنده وابسته به دستوراتی است که در بدنه حلقه شمارنده را مقداردهی می کنند و ممکن است در هر تکرار حلقه وابسته به مسیری که حلقه اجرا می گردد، متفاوت باشد. در بدنه یک حلقه تکرار، مسیرهای مختلفی وجود دارد که در هر تکرار، حلقه از یکی از این مسیرها، مسیر تکرار، اجرا می گردد و در نهایت از یکی از این مسیرها، مسیر پایان، اجرای خود را خاتمه می دهد. در حقیقت دستورات انشعاب شرطی در بدنه حلقه تکرار سبب ایجاد مسیرهای اجرایی متفاوت می گردد. انتخاب دستور بعد از دستورات انشعاب شرطی وابسته به مقدار عبارت شرطی این دستورات خواهد بود. هر عبارت شرطی ترکیب فصلی یا عطفی عبارت رابطه ای می باشد که عملوندهای یک عبارت رابطه ای می توانند متغیرها و ثوابت باشند. بنابراین مقدار عبارت شرطی یک دستور انشعاب شرطی،  $bc$ ، به مقدار متغیرها،  $v$ ، در نقطه ای از برنامه،  $w$ ، که دستور بکار رفته وابسته می باشد و با  $bc(v, w)$  آنرا می توان نمایش داد. بنابراین اگر حلقه بخواهد از یک مسیر خاص اجرا گردد باید متغیرها مقداری داشته باشند

که بعضی از این نقاط هرگز انتخاب نخواهند شد و تخمین حاضر از حداکثر تکرار یک بیش تخمین<sup>۴</sup> خواهد بود.

برای اینکه مقدار تخمینی برای تعداد تکرار هر مسیر به مقدار واقعی آن نزدیک تر شود، تلاش می گردد که فضای  $n$  بعدی کاهش پیدا کند. عبارت دیگر تصویری از چند وجهی در فضای  $m$  بعدی ( $m < n$ ) را بدست می آوریم (شکل ۱). واضح است نقاط درون چند وجهی بدست آمده به مراتب کمتر از چند وجهی اصلی خواهد بود. بنابراین برای تخمین تعداد تکرار هر مسیر مراحل زیر باید انجام گردد:

با استفاده از یک حل کننده محدودیت [7] شرط مسیر را تست کرده و اگر شرط به نادرست ارزیابی گردید، تعداد تکرار آنها صفر لحاظ می گردد. بدین ترتیب مسیرهای اجرا نشدنی<sup>۵</sup> شناسایی می گردد که در نزدیک بودن تخمین به مقدار واقعی بیشترین تعداد تکرار تاثیر بسزایی دارد [1].

اگر مقدار یک متغیر کنترلی در مسیر تغییری نداشت یعنی  $\Delta cv = 0$  آنگاه تصویر چندوجهی بر روی سایر ابعاد بجز این بعد باید محاسبه نمود (قسمت (ب) شکل شماره ۱).

تصویر چند وجهی را برای هر بُعد بدست می آوریم. نتیجه حاصل یک پاره خط خواهد بود که بر اساس آن مقدار شروع  $V_s$  و مقدار پایانی  $V_e$  متغیر کنترلی متناظر بُعد بدست می آید (قسمت (ج) شکل شماره ۱). سپس با استفاده از تابع نحوه تغییر متغیر کنترلی در عبارت (۲) برای هر متغیر کنترلی مقدار  $f_{cv}$  را محاسبه می نماییم [8]:

$$\text{اگر } V_e > V_s \text{ و } a > 1 :$$

$$f_{cv} = \left\lceil \log_a \left( 1 + \frac{(V_e - V_s) * (a - 1)}{b + (a - 1) * V_s} \right) \right\rceil + 1$$

$$\text{اگر } V_e > V_s \text{ و } a = 1 :$$

$$f_{cv} = \left\lceil \frac{V_e - V_s}{b} \right\rceil + 1 \quad (3)$$

برای مسیر تکرار  $ip$  و متغیرهای کنترلی وابسته به آن  $CV$  می توان تعداد تکرار را از رابطه زیر بدست آورد:

$$P_{ip} = \text{Max}_{cv \in CV} f_{cv}$$

حال محدودیت  $X_{ip} < P_{ip}$  به محدودیتهای مسئله برنامه ریزی خطی اضافه می کرده و مراحل فوق را برای سایر مسیرهای تکرار مجدداً انجام می گردد.

#### ۵- کارهای مرتبط

هیلی و همکارانش [9] با استفاده از تحلیل جریان داده تعداد تکرار حلقه‌هایی که منطبق بر الگوی ارائه شده محاسبه می نمایند. در این روش مقدار ابتدایی و انتهایی شمارنده باید یک ثابت باشد و شمارنده فقط می تواند با یک مقدار ثابت جمع یا تفریق گردد. مقدار بدست آمده برای کل بدنه حلقه تکرار می باشد. در شرطها استفاده از عملگرهای منطقی نباید استفاده گردد. ابزار  $aiT$  با استفاده از تحلیل جریان داده ها نامتغیرهای حلقه را استخراج می نماید [10,11]. شمارنده فقط با یک مقدار ثابت جمع یا تفریق می گردد. در مسیرهای مختلف نحوه تغییر شمارنده می تواند

مسیر تکرار  $i$  و  $T_{ip}$  زمان اجرای دنباله دستورات روی یکی از مسیرهای پایان می باشد.  $X_i$  تعداد دفعات اجرا شده مسیر تکرار نام می باشد. IP مجموعه کلیه مسیرهای تکرار می باشد. در این مقاله فرض شده است که  $T_i, T_s$  و  $T_t$  قبلاً توسط تحلیل سطح پایین [1] محاسبه شده است. بنابراین باید  $X_i$  ها به نحوی انتخاب گردند که  $T_{loop}$  مقدار بیشینه داشته باشد و خواهیم داشت

$$WCET_{loop} = WCET_s + \text{Max} \left( \sum_{i \in IP} WCET_i * X_i \right) + \text{Max}_{t \in TP} (WCET_{tp}) \quad (1)$$

$WCET$  بیشترین زمان اجرای دنباله ای از دستورات بر روی یک مسیر می باشد که معلوم فرض شده است باید در جمله دوم (۱)  $X_i$  به نحوی انتخاب گردد که حاصل کل جمله دوم بیشینه گردد. این یک مسئله برنامه ریزی خطی صحیح می باشد که باید مقدار  $X_i$  ها به نحوی انتخاب گردد که تابع هزینه آن  $\sum_{i \in IP} WCET_i * X_i$  روی تعدادی محدودیت  $X_i \leq P_i$  بیشینه گردد. در بخش بعدی نحوه تخمین  $P_i$  شرح داده شده است.

#### ۴- تخمین بیشترین تکرار برای هر مسیر

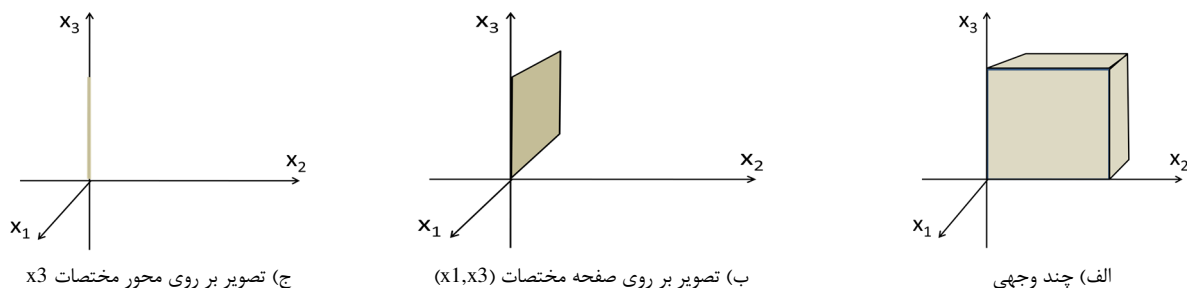
در بیشتر حلقه های تکرار عبارات شرطی (bc) یک نامعادله خطی می باشد. مجموعه جوابهای یک نامعادله خطی برابر یک نیم فضا از فضای  $n$  بعدی است که توسط معادله خطی متناظر با نامعادله تعریف می گردد.  $n$  تعداد متغیرهای کنترلی حلقه می باشد. یک شرط مسیر تکرار یا پایان یک دستگاه نامعادلات خطی از عبارات شرطی می باشد. پاسخ این دستگاه نامعادلات خطی تقاطع (اشتراک) نیم فضاهای تعریف شده توسط هر یک از نامعادلات خطی مربوط به عبارت شرطی می باشد. اگر هر یک از نیم فضاها محدب باشد پاسخ دستگاه که اشتراک این نیم فضاها می باشد یک چندوجهی محدب خواهد بود. این چندوجهی میتواند بسته یا باز باشد. کاهش یا افزایش متغیرهای کنترلی نیز در اغلب حلقه ها یک تابع خطی از متغیر کنترلی می باشد. مثلاً برای متغیر کنترلی  $cv$  نحوه تغییر آن می تواند به صورت زیر باشد:

$$cv_{i+1} = a * cv_i + b \quad (2)$$

$cv_i$  مقدار متغیر در تکرار نام و  $cv_{i+1}$  مقدار متغیر در تکرار  $i+1$  می باشد. همچنین مقدار تغییر متغیر کنترلی بدین صورت تعریف می گردد:

$$\Delta cv = cv_{i+1} - cv_i$$

برای حلقه هایی که شرط مسیر تکرار یک چند وجهی بسته باشد، هر نقطه درون این چند وجهی یک پاسخ برای دستگاه نامعادلات نشان دهنده شرط تکرار می باشد. به عبارتی دیگر هر پاسخ دستگاه معادل چندتایی از مقدار متغیرهای کنترلی خواهد بود که عبارت شرط مسیر را به درست ارزیابی می کند. بنابراین تعداد نقاط درون چند وجهی برابر تعداد تمام حالات مقداردهی متغیرهای کنترلی خواهد بود. پس با محاسبه تعداد این پاسخ ها می توان حداکثر تعداد که حلقه می تواند از این مسیر اجرا گردد را محاسبه نمود [6]. با توجه به نحوه تغییر متغیرهای کنترلی واضح است



شکل (1): چندوجهی و تصویر آن

در بدنه حلقه تکرار چهار مسیر تکرار وجود دارد:

ip<sub>1</sub> = C D E F G L M N  
 ip<sub>2</sub> = C D E F G L N  
 ip<sub>3</sub> = C D H K L M N  
 ip<sub>4</sub> = C D H K L N

همچنین یک مسیر پایان tp = CK وجود دارد.

شرط تکرار و خاتمه برای هر یک از مسیرهای فوق در قالب یک دستگاه نامعادلات خطی و نحوه تغییر شمارنده در هر مسیر بصورت زیر می‌باشد:

$$\text{ipCondition}(ip_1) = \begin{cases} i \geq 1 \\ i < 10 \\ i < j \\ i + 1 > 5 \\ j \leq 11 \end{cases} \quad \text{chgVal}(ip_1): i = i + 2, j = j - 2$$

$$\text{ipCondition}(ip_2) = \begin{cases} i \geq 1 \\ i < 10 \\ i < j \\ i + 1 \leq 5 \\ j \leq 11 \end{cases} \quad \text{chgVal}(ip_2): i = i + 1, j = j - 2$$

$$\text{ipCondition}(ip_3) = \begin{cases} i \geq 1 \\ i < 10 \\ i > j \\ i + 1 > 5 \\ j \leq 11 \end{cases} \quad \text{chgVal}(ip_3): i = i + 2$$

$$\text{ipCondition}(ip_4) = \begin{cases} i \geq 1 \\ i < 10 \\ i > j \\ i + 1 \leq 5 \\ j \leq 11 \end{cases} \quad \text{chgVal}(ip_4): i = i + 1$$

در عبارات فوق آیت  $i > 1$  و  $j \leq 11$  بر اساس مقدار اولیه شان و همچنین افزایش/کاهش یکنواخت در بدنه حلقه به هر شرط مسیر اضافه گردیده است. نکته دیگر که باید مورد توجه قرار گیرد عبارت شرط  $i < 5$  بر اساس مقدار  $i$  در دستور  $L$  می‌باشد. بنابراین این شرط به  $i + 1 < 5$  تغییر پیدا نموده که بر اساس مقدار  $i$  در ابتدای حلقه تکرار خواهد بود.

مسیر تکرار ip<sub>1</sub> یک مسیر اجرا نشدنی نمی‌باشد و همچنین هر دو متغیر کنترلی روی این مسیر تغییر می‌کند. بنابراین تصویر چند وجهی بر روی هر یک از ابعاد به صورت زیر خواهد بود:

تصویر بر محور  $i$ :  $i > 4$  and  $i < 10$

متفاوت باشد. در ابزار TuBound یک معادله سمبولیک ایجاد نموده و با تعدادی قانون آنرا حل می‌نماید. تغییر شمارنده در بدنه حلقه تکرار در این روش مجاز نمی‌باشد [12]. در مرجع [13] روشی ارائه شده که برای حلقه های ساده (یک مسیر در بدنه حلقه) یک معادله بازگشتی با استفاده از تحلیل نمادین ایجاد می‌نماید. سپس با حل این معادله تعداد تکرار را محاسبه می‌نماید. برای حلقه هایی با چند مسیر آنرا به حلقه با یک مسیر تبدیل می‌نماید. همچنین باید تغییر شمارنده یکنواخت باشد.

در روش ارائه شده در این مقاله، حلقه ها با چند مسیر و چند شمارنده می‌توانند تحلیل گردند. محدودیتی در محل دستورات شرط و تغییر دهنده شمارنده و تعدد آن وجود ندارد. شرطها باید خطی باشند و شرط تکرار برای هر مسیر باید یک چند وجهی بسته باشد. همچنین تغییر متغیر کنترلی باید یکنواخت باشد: ممکن است تابع تغییر متغیرهای کنترلی در مسیر فعلی تکرار سبب شود که مسیر اجرای تکرار بعدی مسیر فعلی نباشد. اگر پس از تکرار نام حلقه از یک مسیر ip<sub>i</sub>، حلقه در تکرار بعدی از مسیر دیگری اجرا گردد و پس از چند تکرار مجدداً در تکرار نام از مسیر ip<sub>i</sub> تکرار گردد، برای حلقه های تکرار افزایشی (کاهشی) مقدار هیچ یک از متغیرهای کنترلی در تکرار نام از مقدار آن متغیر در تکرار نام کمتر (بیشتر) نباشد.

## ۶- مثال

در شکل شماره ۲ یک قطعه کد دارای یک حلقه while نشان داده شده است.

```

A   i = 1;
B   j = 11;
C   while(i < 10) {
D       if(i < j) {
E           i = i + 1;
F           j = j - 2;
G       }
H   else
K       i = i + 1;
L   if(i > 5)
M       i = i + 1;
N   }
P
    
```

شکل (۲): قطعه کد برای تخمین بیشترین زمان اجرای حلقه

$$WCET_{loop} = 20 + 60X_{ip1} + 50X_{ip2} + 50X_{ip3} + 40X_{ip2} + 10$$

محدودیتها عبارتند از  $(X_{ip} \leq P_{ip})$ :

$$X_{ip1} \leq 3, X_{ip2} \leq 5, X_{ip3} \leq 3, X_{ip4} \leq 4$$

با حل مسئله برنامه ریزی خطی صحیح فوق داریم:

$$WCET_{loop} = 770$$

### ۷- ارزیابی

برای ارزیابی حلقه‌هایی از برنامه‌های محک Malardalen [14] و مثالهای موجود در گزارشات سایر پژوهشها انتخاب گردید تا توانایی روش ارائه شده در این مقاله نشان داده شود. این قطعات کد در \* نشان داده شده اند. در این بخش مشخصاً روش ارائه شده با روش [13] مقایسه گردیده است. البته این روش پیاده‌سازی نشده است و نتایج ارائه شده برای روش [13] به صورت دستی بدست آمده است. همچنین فرض شده است که زمان اجرای همه دستورات در بدنه حلقه ثابت و معلوم است و بیشترین زمان اجرای دستورات قبل از شروع تکرار (WCETs) و همچنین بیشترین زمان اجرای دستورات بر روی مسیر خاتمه (WCETip) در رابطه (۱)

تصویر بر محور  $j$ :  $j > 5$  and  $j \leq 11$

که بیشترین مقدار ممکن برای  $i$  مقدار ۹ و کمترین مقدار ۵ خواهد بود. همچنین برای  $j$  بیشترین مقدار ۱۱ و کمترین مقدار ۶ خواهد بود. توجه به (۳) می توان تعداد تکرار از این مسیر را محاسبه نمود:

$$Pip_1 = \text{Max}(\lfloor (9-5)/2 \rfloor + 1, \lfloor (11-6)/2 \rfloor + 1) = 3$$

بهمین صورت برای مسیر تکرار  $ip_2$  داریم:

$$Pip_2 = \text{Max}(\lfloor (4-1)/1 \rfloor + 1, \lfloor (11-2)/2 \rfloor + 1) = 5$$

در دو مسیر تکرار بعدی مقدار  $j$  تغییر نمی کند. بنابراین باید تصویر چندوجهی نشان دهنده دستگاه نامعادلات خطی آنها بر روی محور  $i$  بدست آورد. سپس تعداد تکرار را بر اساس (۳) محاسبه نمود.

$$Pip_3 = \lfloor (9-5)/2 \rfloor + 1 = 3$$

$$Pip_4 = \lfloor (4-1)/1 \rfloor + 1 = 4$$

با فرض اینکه زمان اجرای هر دستور در قطعه کد فوق ۱۰ واحد زمانی باشد، خواهیم داشت:

$$T_s = 20, T_{ip} = 10$$

$$Tip_1 = 60, Tip_2 = 50, Tip_3 = 50, Tip_4 = 40$$

با توجه به (۱) باید مقدار تابع هزینه زیر بیشینه گردد:

جدول ۱- قطعات کد نمونه برای ارزیابی روش مبتنی بر انطباق الگو

نام قطعه کد	قطعه کد برنامه	نام برنامه محک و منبع
مثال ۱	<pre>for( i=0 ; i&lt;10;i++){     a = IN ;     if(i&lt;5) {         a = a &amp; 0x0F ;         OUT = num_to_lcd(a) ;     } }</pre>	[14] و <i>lcdnum</i>
مثال ۲	<pre>for(j=0 ; j&lt;100; j++) {     if( a[k][j] == 1) j++;     else a[k][j] = -1; }</pre>	شکل ۱ در [13]
مثال ۳	<pre>while ( k &lt; j ) {     j *= k ;     k /= 2 ; }</pre>	<i>fft1</i> در [14]
مثال ۴	<pre>for( i=1 ; i&lt;=100 ; i++){     a = -i * (nm1 + i) ;     b += 2 ;     d = 10 * ( a *d + b ) ;     c = b + a / c ;     del = c*d ;     h *= del ;     if( del &lt; 10000) {         ans = h * -x;         return ans;     } }</pre>	<i>expint</i> در [14]
مثال ۵	<pre>b=5; for( i=1 ; i&lt;=100 ; i++){     b += 2 ;     c=5* i+b ;     if ( c &gt; 42) {         break ;     } }</pre>	مثال ارائه شده در این مقاله

محدودیت‌هایی به مسئله برنامه ریزی خطی اضافه گردد تا دقت روش ارائه شده افزایش یابد.

## مراجع

- [1] Wilhelm R., Engblom J., Ermedahl A., Holsti N., Thesing S., Whalley D., Bernat G., Ferdinand C., Heckmann R., Mitra T., Mueller F., Puaut I., Puschner P., Staschulat J., Stenström P. *The worst-case execution time problem – overview of approaches and survey of tools*. Trans. on Embedded Computing Sys., 7(3):1-53, 2008.
- [2] Chapman R. *Static Timing Analysis and Program Proof*, PhD dissertation, Univ. of York, 1995.
- [3] Gustafsson J., Ermedahl A., Sandberg C., Lisper B. *Automatic derivation of loop bounds and infeasible paths for WCET analysis using abstract execution*. In Proc. 27th IEEE Real-Time Systems Symposium, 2006.
- [4] Michiel M., Bonenfant A., Cassé H., Sainrat P. *Static loop bound analysis of C programs based on flow analysis and abstract interpretation*. In: IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), pages 161-166, Taiwan, 2008.
- [5] Saeed Parsa, Mehdi Sakhaei-nia, *A New Approach to Determine Number of Loop Iterations for WCET Analysis*. ITNG 2011: 760-765.
- [6] Bygde S., Ermedahl A., Lisper B., *An efficient algorithm for parametric WCET calculation*, Journal of Systems Architecture Volume 57, Issue 6, June 2011, Pages 614-624
- [7] Moura L. M. de , Bjørner. N. , *Z3: An efficient SMT solver*. In TACAS, volume 4963 of LNCS, pages 337-340. Springer, 2008.
- [8] Djemai Kebbal, Pascal Sainrat *Combining Symbolic Execution and Path Enumeration in Worst-Case Execution Time Analysis*. In Proc. 6<sup>th</sup> International Workshop on Worst-Case Execution Time Analysis, (WCET'2006), July 2006
- [9] Healy C., Sjodin M., Rustagi V., Whalley D., Engelen R. *Supporting timing analysis by automatic bounding of loop iterations*. Journal of Real-Time Systems, 18(2-3):129-156, 2000.
- [10] Cullmann C., Martin F. *Data-flow based detection of loop bounds*. In 7th International Workshop on Worst-Case Execution Time Analysis, (WCET'2007), Italy, July 2007.
- [11] aiT Tool( 2007) <http://www.absint.com>.
- [12] Prantl A., Schordan M., Knoop J. *TuBound-A Conceptually New Tool for Worst-Case Execution Time Analysis*, In Proceedings 8th Int. Workshop on Worst-Case Execution Time Analysis (WCET 2008), 141-148, Prague, Czech Republic, 2008.
- [13] Knoop J. , Kovács L., and Zwirchmayr J. *Symbolic Loop Bound Computation for WCET Analysis*. Proc. PSI, Ershov Informatics Conference June, 27 -July, 1, 2011, Novosibirsk, Russia, 2011.
- [14] Gustafsson J., Betts A., Ermedahl, A., Lisper B. *The mälardalen WCET benchmarks: Past, present and future*, in *10th International Workshop on Worst-Case Execution Time Analysis (WCET 2010)*, 2010.

## زیر نویس‌ها

<sup>1</sup> deadline

<sup>2</sup> WCET: Worst-Case Execution Time

<sup>3</sup> Pattern-matching

<sup>4</sup> overestimation

<sup>5</sup> Infeasible path

قابل ملاحظه نبوده و در محاسبات نادیده گرفته شده است. مثال شماره یک توسط روش ارائه شده در این مقاله نسبت به [13] دقیق‌تر تخمین زده می‌شود چون در [13] دستور شرط در اجرای حلقه نادیده گرفته می‌شود و تعداد تکرار دستورات در بدنه شرط را برابر با تعداد تکرار حلقه در نظر می‌گیرد. در مثال ۲ مقدار تخمینی در دو روش یکسان می‌باشد. مثال ۳ توسط هر دو روش قابل تخمین نمی‌باشد. در روش ارائه شده در این مقاله تصویر چندوجهی پاره خط نبوده که بتوان تخمینی را ارائه نماید، اما در صورتیکه عبارت  $k \geq 0$  را بتوان به شرط تکرار اضافه نمود، این روش می‌تواند تخمینی از بیشترین زمان اجرای حلقه را ارائه نماید. مثال ۴ یک حلقه می‌باشد که اجرای آن با یک انشعاب شرطی در بدنه حلقه خاتمه می‌پذیرد. به عبارت دیگر، یک حلقه چند مسیری بوده با خاتمه ناگهانی (Abrupt termination) می‌باشد. روش [13] انشعاب شرطی را نادیده گرفته و تعداد تکرار را برای این حلقه ۱۰۰ تکرار تخمین می‌زند. روش ارائه شده در این رساله همین تخمین را ارائه می‌نماید. چون عبارت شرطی غیر خطی بوده از عبارت شرط تکرار مسیر حذف می‌گردد و تعداد تکرار ۱۰۰ تخمین زده خواهد شد. اما اگر مانند مثال ۵، عبارت شرط خطی می‌بود، روش [13] مجدداً همان مقدار قبلی را تخمین می‌زد. اما روش ارائه شده در این مقاله تعداد تکرار این حلقه را ۷ تکرار تخمین می‌زند که بسیار دقیق‌تر می‌باشد. همچنین در مثال ۵ نشان داده شده است که اگر یک حلقه چند مسیری با بیش از یک متغیر کنترلی حلقه داشته باشیم در روش ارائه شده در این رساله قابل شناسایی بوده و اثرات آن بر تعداد تکرار حلقه لحاظ می‌گردد.

## ۸- نتیجه گیری و کارهای آینده

تخمین بیشترین زمان اجرای حلقه‌های تکرار یکی از مهمترین مسائل تحلیل بیشترین زمان اجرای وظیفه‌ها در نرم افزارهای بی درنگ می‌باشد. در روشهای تطبیق الگو برای محاسبه تعداد تکرار، ساختار و مکان شرطهای تست کننده شمارنده و مکان و چگونگی تغییر شمارنده در بدنه حلقه از مسائلی می‌باشند که سبب عدم تطبیق الگو و نتیجتاً شکست محاسبه تعداد تکرار می‌گردد. در این مقاله یک روش نوین برای حل مشکل ارائه گردیده است که کمترین وابستگی را به شرطها و دستورات تغییر دهنده شمارنده در بدنه حلقه تکرار دارد. روش ارائه شده مسیر اجرایی در بدنه حلقه تکرار که در هر تکرار حلقه از آنها اجرا می‌گردد، را شناسایی می‌نماید. سپس برای هر مسیر عبارت شرط مسیر و همچنین نحوه تغییر شمارنده بر روی این مسیر را ایجاد می‌نماید. سپس با استفاده از این دو دسته عبارات تعداد تکرار هر مسیر را تخمین می‌زند. در صورتیکه بیشترین زمان اجرای ممکن برای هر مسیر را داشته باشیم می‌توانیم بیشترین زمان اجرای حلقه تکرار را تخمین بزنیم.

در کارهای آینده سعی می‌گردد که برای چندوجهی باز نشان دهنده شرط مسیر و تغییر غیر یکنواخت شمارنده، روش ارائه شده توسعه یابد. همچنین بر اساس مقدار متغیرهای کنترلی وابستگی بین مسیرها شناسایی و

# SID



سرویس های ویژه



سرویس ترجمه تخصصی



کارگاه های آموزشی



بلاگ مرکز اطلاعات علمی



سامانه ویراستاری STES



فیلم های آموزشی

## کارگاه های آموزشی مرکز اطلاعات علمی جهاد دانشگاهی

کارگاه آنلاین  
بررسی مقابله ای متون (مقدماتی)

کارگاه آنلاین  
پروپوزال نویسی و پایان نامه نویسی

کارگاه آنلاین آشنایی با پایگاه های اطلاعات علمی بین المللی و ترند های جستجو  
بین المللی و ترند های جستجو