دانشگاه تبریز

دانشکده علوم ریاضی

گروه علوم کامپیوتر

پایان‌نامه

برای دریافت درجه کارشناسی ارشد

در رشته علوم کامپیوتر گرایش سیستم‌های هوشمند

عنوان

ارائه یک رویکرد تکاملی برای خوشه‌بندی نرم‌افزار جهت اولویت‌بندی خوشه‌ها در آزمون نرم‌افزار

استاد راهنما

دکتر شهریار لطفی

استاد مشاور

دکتر جابر کریم‌پور ینگجه

پژوهشگر

پریسا محمدزاده وردین

بهمن ۱۳۹۳

بسم الله الرحمن الرحيم

ماحصل آموخته‌هایم را تقدیم می‌کنم به آنان که مهر آسمانی‌شان آرام‌بخش آلام زمینی‌ام است

به استوارترین تکیه‌گاهم، دستان پرمهر پدرم

به سبزترین نگاه زندگیم، چشمان سبز مادرم

که هرچه آموختم در مکتب عشق شما آموختم و هرچه بکوشم قطره‌ای از دریای بی‌کران مهربانیتان را سپاس نتوانم بگویم.

امروز هستی‌ام به امید شماست و فردا کلید باغ بهشتم رضای شما

راه آوردی گران‌سنگ‌تر از این ارزان نداشتم تا به خاک پایتان نثار کنم، باشد که حاصل تلاشم نسیم گونه غبار خستگیتان را بزداید.

بوسه بر دستان پرمهرتان

تشکر و قدردانی

نخستین سپاس و ستایش از آن خداوندی است که بنده کوچکش را در دریای بیکران اندیشه، قطره‌ای ساخت تا وسعت آن را از دریچه‌ی اندیشه‌های ناب آموزگارانی بزرگ به تماشا نشیند. لذا اکنون که در سایه سار بنده نوازی‌هایش پایان نامه ی حاضر، به انجام رسیده است، بر خود لازم می دانم تا مراتب سپاس را از بزرگوارانی به جا آورم که اگر دست یاری گرشان نبود، هرگز این پایان نامه به انجام نمی رسید.

ابتدا از استاد عالی قدرم جناب آقای دکتر شهریار لطفی که با نکته‌های دلاویز و گفته‌های بلند، صحیفه‌های سخن را علم پرور نمود و همواره راهنما و راه گشای نگارنده در اتمام و اکمال پایان نامه بوده است، کمال سپاس را دارم.

از استاد گرانقدرم جناب آقای دکتر جابر کریم پور که زحمت مشاوره این پایان نامه را متحمل شدند، صمیمانه تشکر می کنم

سپاس آخر را به مهربانترین همراهان زندگیم، به پدر و مادر عزیزم تقدیم می کنم که حضورشان در فضای زندگیم مصداق بی ریای سخاوت بوده است.

پریسا محمدزاده وردین

بهمن ۱۳۹۳

نام خانوادگی دانشجو: محمدزاده وردین **نام:** پریسا

عنوان پایان‌نامه: ارائه یک رویکرد تکاملی برای خوشه‌بندی نرم‌افزار جهت اولویت‌بندی خوشه‌ها در آزمون نرم‌افزار

استاد راهنما: دکتر شهریار لطفی

استاد مشاور: دکتر جابر کریم‌پور ینگجه

مقطع تحصیلی: کارشناسی ارشد **رشته:** علوم کامپیوتر **گرایش:** سیستم‌های هوشمند **دانشگاه:** تبریز

دانشکده: علوم ریاضی **تاریخ فارغ التحصیلی:** بهمن ۱۳۹۳ **تعداد صفحه:** ۱۴۱

کلید واژه‌ها: آزمون نرم‌افزار، خوشه‌بندی، اولویت‌بندی و الگوریتم رقابت استعماری

چکیده:

نیاز و وابستگی روز افزون شرکت‌ها و سازمان‌ها به رایانه و استفاده از آن باعث گردیده تا مدیران و کارآفرینان با استفاده از نرم‌افزارهای مختلف به دنبال کارآیی هر چه بهتر مجموعه‌ی خود باشند. امروزه نرم‌افزارها در بسیاری از کاربردهای بحرانی نیز نقش مهمی دارند به طوری‌که یک خطا در نرم‌افزار ممکن است باعث بروز خسارت‌های غیر قابل جبران گردد. در نتیجه نرم‌افزار باید تا حد امکان عاری از هرگونه خطا و خرابی باشد. یکی از مهمترین دغدغه‌های مهندسان نرم‌افزار ارائه‌ی نرم‌افزار کامل، بی نقص و مطابق با نیازمندی‌های مشتری‌ها است که در قسمت آزمون نرم‌افزار همه‌ی این موارد قبل از تحویل نهایی، بررسی می‌شود. بر اساس دلیل‌های ذکر شده، آزمون نرم‌افزار بخش مهم و زمان‌بری از فرآیند تولید نرم‌افزار را در بر می‌گیرد. از طرفی در برخی موارد، آزمون کامل نرم‌افزار امکان‌پذیر نمی‌باشد؛ زیرا با توجه به صبر مشتری ممکن است زمان کافی برای آزمودن کامل نرم‌افزار وجود نداشته باشد و دیرکرد تحویل نرم‌افزار موجب زیان‌های مالی بسیاری شود. به همین علت کمینه‌سازی زمان آزمون نرم‌افزار از جمله اهداف آزمون است.

در این پایان‌نامه هدف ارائه‌ی الگوریتمی برای خوشه‌بندی نرم‌افزار و اولویت‌بندی این خوشه‌ها در جهت

آزمون نرم‌افزار است. در الگوریتم پیشنهادی در راستای رسیدن به هدف کمینه‌سازی زمان آزمون، از خوشه‌بندی و اولویت‌بندی استفاده شده است. به این ترتیب که ابتدا با استفاده از الگوریتم رقابت استعماری، گراف وابستگی نرم‌افزار، خوشه‌بندی می‌شود. این خوشه‌بندی بر اساس افزایش اتصال درون خوشه و کاهش اتصال برون خوشه‌ای انجام می‌گیرد. سپس به وسیله‌ی معیارهایی در زمینه‌ی تشخیص مناطق مستعد خرابی، این خوشه‌ها اولویت‌بندی می‌شوند. حال اگر آزمون‌گر نرم‌افزار به جای آزمودن کل برنامه به صورت یکپارچه، این خوشه‌ها را به ترتیب اولویت بیازماید، این کار سبب کاهش زمان آزمون شده و در کمترین زمان ممکن، بیشترین تعداد خطا به دست می‌آید. راه‌کار ارائه شده بر روی مجموعه برنامه‌های زیمنس اعمال شده است. نتایج حاصل از این ارزیابی نشان می‌دهد که خوشه‌های با اولویت بالاتر حامل خطاها و خرابی‌های بیشتری هستند.

فهرست مطالب

عنوان **صفحه**

فصل اول مقدمه ۱

۱-۱ اصطلاحات ۳

۱-۲ بیان مسئله ۴

۱-۳ اهداف پایان‌نامه ۵

۱-۴ نظریه ۶

۱-۵ سازمان پایان‌نامه ۶

فصل دوم مفاهیم پایه‌ای و راه‌کارهای گذشته ۷

۲-۱ مفاهیم پایه‌ای ۸

۲-۱-۱ آزمون نرم‌افزار ۸

۲-۱-۲ اهداف آزمون نرم‌افزار ۹

۲-۱-۳ آزمون جعبه سفید و جعبه سیاه ۹

2-1-4 آزمون رگرسیون ۹

۲-۱-۵ طراحی نمونه‌های آزمون ۱۰

۲-۱-۶ اشکال‌زدایی ۱۰

۲-۱-۷ خوشه‌بندی ۱۰

۲-۱-۸ معیارهای تشابه ۱۱

۲-۱-۹ روش‌های خوشه‌بندی ۱۴

۲-۱-۱۰ الگوریتم ژنتیک ۱۹

۲-۱-۱۱ عملگرهای الگوریتم ژنتیک ۲۰

۲-۱-۱۲ الگوریتم رقابت استعماری ۲۱

۲-۱-۱۳ سیاست‌های الگوریتم رقابت استعماری ۲۲

۲-۲ تولید نمونه آزمون مناسب ۲۵

۲-۲-۱ کاربرد الگوریتم ژنتیک در تولید نمونه آزمون ۲۵

۲-۳ کاهش نمونه آزمون‌ها ۲۹

۲-۳-۱ بهینه‌سازی و تسریع فرآیند کشف خطا در آزمون رگرسیون نرم‌افزار ۳۰

۲-۳-۲ مسئله آزمون رگرسیون کاهشی زمان آگاه ۳۳

۲-۴ اولویت‌بندی نمونه آزمون‌ها ۳۶
۲-۴-۱ اولویت‌بندی نمونه آزمون‌ها بر اساس نرخ کشف خرابی ۳۶
۲-۴-۲ اولویت‌بندی نمونه آزمون‌ها بر اساس وابستگی خرابی‌ها ۴۰
۲-۴-۳ اولویت‌بندی نمونه آزمون‌ها با استفاده از خوشه‌بندی ۴۵
۲-۴-۴ اولویت‌بندی خوشه‌های نمونه آزمون‌ها ۴۹
۲-۵ اشکال‌زدایی ۵۳
۲-۵-۱ تشخیص اجراهای خطادار نرم‌افزار با استفاده از راه‌کار خوشه‌بندی ۵۴
۲-۵-۲ مکان‌یابی خطاهای چندگانه در برنامه ۵۷
۲-۶ خوشه‌بندی ۶۱
۲-۶-۱ خوشه‌بندی خودکار سامانه‌های نرم‌افزاری به وسیله الگوریتم ژنتیک ۶۲
۲-۶-۲ خوشه‌بندی نرم‌افزار به وسیله‌ی الگوریتم ترکیبی ۶۵
۲-۶-۳ خوشه‌بندی نرم‌افزار به وسیله‌ی الگوریتم ترکیبی وزن‌دار ۶۹
۲-۷ مقایسه روش‌های آزمون نرم‌افزار ۷۳
۲-۸ خلاصه فصل ۷۴
فصل سوم راه‌کار پیشنهادی و نتایج علمی ۷۵
۳-۱ راه‌کار پیشنهادی ۷۷
۳-۱-۱ تشریح راه‌کار پیشنهادی ۷۷
۳-۱-۲ خوشه‌بندی ۷۹
۳-۱-۳ اولویت‌بندی ۹۰
۳-۲ ارزیابی نتایج ۹۴
۳-۲-۱ محیط آزمایش ۹۵
۳-۲-۲ پارامترهای الگوریتم پیشنهادی ۹۵
۳-۲-۳ ارزیابی الگوریتم خوشه‌بندی ۹۶
۳-۲-۴ ارزیابی اولویت‌بندی ۱۰۷
۳-۲-۵ ارزیابی یکپارچه الگوریتم ارائه شده ۱۱۰
۳-۳ بحث ۱۳۱
۳-۴ خلاصه فصل ۱۳۲
فصل چهارم نتیجه‌گیری و راه‌کارهای آتی ۱۳۴

۴-۱ در اثبات نظریه ۱۳۵
۴-۲ در تحقق اهداف پایان‌نامه..... ۱۳۵
۴-۳ کارهای مرتبط، بحث و مقایسه ۱۳۶
۴-۴ دستاوردها..... ۱۳۶
۴-۵ موضوعات پژوهشی آینده ۱۳۷
مراجع ۱۳۸

فهرست جدول‌ها

عنوان **صفحه**

جدول ۲-۱: جدول‌های انجام محاسبات تا رسیدن به جواب بهینه ۲۹

جدول ۲-۲: تعداد خرابی‌های کشف شده به وسیله‌ی هر نمونه آزمون ۳۹

جدول ۲-۳: ماتریس وابستگی ۴۲

جدول ۲-۴: خرابی‌های کشف شده به وسیله‌ی هر نمونه آزمون ۴۲

جدول ۲-۵: مسیرهای مستقل برای شبه کد ذکر شده ۵۲

جدول ۲-۶: موجودیت‌ها و بردار ویژگی آنها ۶۸

جدول ۲-۷: میزان تشابه موجودیت‌ها با معیار جاکارد ۶۸

جدول ۲-۸: رابطه موجودیت‌ها با ویژگی‌ها ۷۰

جدول ۲-۹: ماتریس تشابه به وسیله معیار جاکارد ۷۱

جدول ۲-۱۰: تعداد موجودیت‌های در ارتباط با ویژگی خاص داخل یک خوشه ۷۱

جدول ۲-۱۱: ماتریس تشابه به وسیله معیار Ellenberg ۷۲

جدول ۲-۱۲: ماتریس تشابه به وسیله معیار Ellenberg اصلاح شده ۷۳

جدول ۲-۱۳: جدول جمع‌بندی ۷۴

جدول ۳-۱: لیست پارامترهای الگوریتم پیشنهادی ۹۶

جدول ۳-۲: نتایج خروجی الگوریتم خوشه‌بندی برای گراف کاترپیلار ۹۸

جدول ۳-۳: نتایج خروجی الگوریتم خوشه‌بندی برای گراف ترکیبی ۱۰۰

جدول ۳-۴: خلاصه نتایج آزمون پایداری الگوریتم خوشه‌بندی بر روی گراف کاترپیلار ۱۰۴

جدول ۳-۵: خلاصه نتایج آزمون پایداری الگوریتم خوشه‌بندی بر روی گراف ترکیبی ۱۰۵

جدول ۳-۶: نتایج الگوریتم خوشه‌بندی بر روی گراف schedule2 ..... ۱۰۷

جدول ۳-۷: مقادیر معیارهای تشخیص خطاپذیری برای گراف schedule2 ..... ۱۰۸
جدول ۳-۸: مقادیر نرمال‌سازی شده‌ی معیارهای تشخیص خطاپذیری برای گراف schedule2 ..... ۱۰۹
جدول ۳-۹: امتیاز گره‌های گراف schedule2 ..... ۱۰۹
جدول ۳-۱۰: خلاصه‌ی نتایج اولویت‌بندی خوشه‌های برنامه‌ی schedule2 ..... ۱۰۹
جدول ۳-۱۱: مشخصات مجموعه برنامه‌های زیمنس ۱۱۰
جدول ۳-۱۲: نتایج حاصل از خروجی الگوریتم پیشنهادی بر روی برنامه‌ی schedule2 ..... ۱۱۲
جدول ۳-۱۳: خلاصه‌ی نتایج آزمون پایداری بر روی برنامه‌ی schedule2 ..... ۱۱۵
جدول ۳-۱۴: لیست شماره گره های اختصاص یافته به گره‌های برنامه schedule ۱۱۷
جدول ۳-۱۵: نتایج حاصل از خروجی الگوریتم پیشنهادی بر روی برنامه‌ی schedule ۱۱۸
جدول ۳-۱۶: خلاصه‌ی نتایج آزمون پایداری برنامه‌ی schedule ۱۲۰
جدول ۳-۱۷: شماره گره اختصاص یافته به گراف برنامه replace ۱۲۲
جدول ۳-۱۸: نتایج حاصل از خروجی الگوریتم پیشنهادی بر روی برنامه‌ی replace ۱۲۳
جدول ۳-۱۹: خلاصه‌ی نتایج آزمون پایداری بر روی برنامه‌ی replace ۱۲۵
جدول ۳-۲۰: شماره اختصاص یافته به گراف برنامه‌ی tcas ۱۲۷
جدول ۳-۲۱: نتایج حاصل از خروجی الگوریتم پیشنهادی بر روی برنامه‌ی tcas ۱۲۸
جدول ۳-۲۲: خلاصه‌ی نتایج آزمون پایداری الگوریتم پیشنهادی در برنامه‌ی tcas ۱۳۰
جدول ۳-۲۳: بررسی میزان درستی عملکرد الگوریتم پیشنهادی در مجموعه زیمنس ۱۳۱
جدول ۳-۲۴: جمع‌بندی کلی از روش‌های پیشین و الگوریتم پیشنهادی ۱۳۳

فهرست شکل‌ها

عنوان **صفحه**

شکل ۲-۱: خوشه‌بندی نمونه‌های ورودی ۱۱

شکل ۲-۲: ماتریس موجودیت و ویژگی‌ها ۱۲

شکل ۲-۳: تفاوت بین روش‌های بالا به پایین با روش‌های پایین به بالا ۱۶

شکل ۲-۴: اجزای اجتماعی-سیاسی تشکیل دهنده‌ی یک کشور ۲۳

شکل ۲-۵: اعمال سیاست جذب در الگوریتم رقابت استعماری ۲۴

شکل ۲-۶: اعمال سیاست انقلاب در الگوریتم رقابت استعماری ۲۴

شکل ۲-۷: کد با گراف CFG وزن‌دار ۲۶

شکل ۲-۸: شبه کد الگوریتم پیشنهادی ۳۸

شکل ۲-۹: گراف وابستگی خرابی‌ها ۴۱

شکل ۲-۱۰: نمودار APFDD برای نمونه آزمون‌های اولویت‌بندی نشده ۴۳

شکل ۲-۱۱: نمودار APFDD برای نمونه آزمون‌های اولویت‌بندی نشده ۴۵

شکل ۲-۱۲: مثال از اولویت‌بندی نمونه آزمون‌ها در خوشه‌ها ۴۷

شکل ۲-۱۳: شبه کد برنامه تشخیص مثلث ۵۱

شکل ۲-۱۴: گراف کنترل جریان برای شبه کد ذکر شده ۵۱

شکل ۲-۱۵: مراحل راه‌کار حاتموند و همکارانش ۵۵

شکل ۲-۱۶: یک نمونه برنامه ساده ۵۶

شکل ۲-۱۷: راه‌کار پیشنهادی ۵۸

شکل ۲-۱۸: الگوریتم همترازی دو دنباله ۵۹

شکل ۲-۱۹: خوشه‌بندی MDG ۶۴

شکل ۲-۲۰: خوشه‌بندی بهتر MDG ۶۴
شکل ۲-۲۱: یک مرحله میانی در طول خوشه‌بندی متراکم شونده ۷۰
شکل ۳-۱: شمای کلی مراحل انجام الگوریتم پیشنهادی ۷۸
شکل ۳-۲: یک نمونه گراف وابستگی..... ۸۰
شکل ۳-۳: یک نمونه خوشه‌بندی برای گراف وابستگی..... ۸۰
شکل ۳-۴: یک نمونه خوشه‌بندی بهتر برای گراف وابستگی..... ۸۰
شکل ۳-۵: یک مثال از کدگذاری..... ۸۲
شکل ۳-۶: نمونه‌ای از جمعیت اولیه..... ۸۲
شکل ۳-۷: یک مثال برای چرخ رولت..... ۸۳
شکل ۳-۸: مثالی از اعمال سیاست جذب ۸۷
شکل ۳-۹: مثالی از اعمال سیاست انقلاب..... ۸۸
شکل ۳-۱۰: مثالی از جابجایی مستعمره و استعمارگر..... ۸۸
شکل ۳-۱۱: شمای کلی الگوریتم رقابت استعماری در جهت خوشه‌بندی نرم‌افزار ۹۰
شکل ۳-۱۲: شمای کلی مراحل اولویت‌بندی خوشه‌ها..... ۹۴
شکل ۳-۱۳: گراف کاترپیلار ۹۷
شکل ۳-۱۴: خوشه‌بندی مناسب برای گراف کاترپیلار..... ۹۸
شکل ۳-۱۵: خروجی الگوریتم خوشه‌بندی برای گراف کاترپیلار..... ۹۸
شکل ۳-۱۶: گراف ترکیبی از سه گراف کامل..... ۹۹
شکل ۳-۱۷: خوشه‌بندی مناسب برای گراف ترکیبی از سه گراف کامل ۹۹
شکل ۳-۱۸: خروجی الگوریتم خوشه‌بندی برای گراف ترکیبی ۱۰۰
شکل ۳-۱۹: بررسی همگرایی الگوریتم خوشه‌بندی به اهداف بهینه‌سازی بر روی گراف کاترپیلار..... ۱۰۱

شکل ۳-۲۰: بررسی همگرایی الگوریتم خوشه‌بندی به اهداف بهینه‌سازی بر روی گراف ترکیبی ۱۰۲
شکل ۳-۲۱: بررسی پایداری الگوریتم خوشه‌بندی به اهداف بهینه‌سازی بر روی گراف کاترپیلار ۱۰۳
شکل ۳-۲۲: بررسی پایداری الگوریتم خوشه‌بندی به اهداف بهینه‌سازی بر روی گراف ترکیبی ۱۰۴
شکل ۳-۲۳: بررسی تعداد استعمارگرها در هر تکرار الگوریتم خوشه‌بندی بر روی گراف کاترپیلار ۱۰۶
شکل ۳-۲۴: بررسی تعداد استعمارگرها در هر تکرار الگوریتم خوشه‌بندی بر روی گراف ترکیبی ۱۰۶
شکل ۳-۲۵: گراف وابستگی برنامه‌ی schedule2 ..... ۱۰۷
شکل ۳-۲۶: خروجی الگوریتم پیشنهادی بر روی برنامه‌ی schedule2 ..... ۱۱۱
شکل ۳-۲۷: بررسی همگرایی الگوریتم پیشنهادی به اهداف بهینه‌سازی بر روی برنامه‌ی schedule2 ..... ۱۱۳
شکل ۳-۲۸: بررسی پایداری الگوریتم پیشنهادی به اهداف بهینه‌سازی بر روی برنامه‌ی schedule2 ..... ۱۱۴
شکل ۳-۲۹: بررسی تعداد استعمارگرها در هر تکرار الگوریتم پیشنهادی بر روی برنامه‌ی schedule2 ..... ۱۱۶
شکل ۳-۳۰: گراف برنامه‌ی schedule ۱۱۷
شکل ۳-۳۱: خروجی الگوریتم پیشنهادی بر روی برنامه‌ی schedule ۱۱۸
شکل ۳-۳۲: بررسی همگرایی الگوریتم پیشنهادی به اهداف بهینه‌سازی بر روی برنامه‌ی schedule ۱۱۹
شکل ۳-۳۳: بررسی پایداری الگوریتم پیشنهادی به اهداف بهینه‌سازی بر روی برنامه‌ی schedule ۱۲۰
شکل ۳-۳۴: بررسی تعداد استعمارگرها در هر تکرار الگوریتم پیشنهادی بر روی برنامه‌ی schedule ۱۲۱
شکل ۳-۳۵: گراف برنامه‌ی replace ۱۲۲
شکل ۳-۳۶: خروجی الگوریتم پیشنهادی بر روی برنامه‌ی replace ۱۲۳
شکل ۳-۳۷: بررسی همگرایی الگوریتم پیشنهادی به اهداف بهینه‌سازی بر روی برنامه‌ی replace ۱۲۴
شکل ۳-۳۸: بررسی پایداری الگوریتم پیشنهادی به اهداف بهینه‌سازی بر روی برنامه‌ی replace ۱۲۵
شکل ۳-۳۹: بررسی تعداد استعمارگرها در هر تکرار الگوریتم پیشنهادی بر روی برنامه‌ی replace ۱۲۶
شکل ۳-۴۰: گراف برنامه‌ی tcas ۱۲۷

شکل ۳-۴۱: خروجی الگوریتم پیشنهادی بر روی برنامه‌ی tcas ۱۲۸
شکل ۳-۴۲: بررسی همگرایی الگوریتم پیشنهادی به اهداف بهینه‌سازی بر روی برنامه‌ی tcas ۱۲۹
شکل ۳-۴۳: بررسی پایداری الگوریتم پیشنهادی به اهداف بهینه‌سازی بر روی برنامه‌ی tcas..... ۱۲۹
شکل ۳-۴۴: بررسی تعداد استعمارگرها در هر تکرار الگوریتم پیشنهادی بر روی برنامه‌ی tcas..... ۱۳۰

فصل اول

مقدمه

از زمان حضور نرم‌افزارهای رایانه‌ای، پیچیدگی و اندازه آنها همواره رو به افزایش است. هر محصول نرم‌افزاری، مخاطبان خاصی دارد؛ بنابراین زمانی که سازمانی یک محصول نرم‌افزاری را می‌نویسد یا خریداری می‌کند، به طور منطقی باید اطمینان حاصل کند که آیا محصول نرم‌افزاری برای کاربران او، مخاطبان او، خریداران و سایر سهام‌داران قابل پذیرش است یا خیر. آزمون نرم‌افزار[1] فرآیند تلاش برای این‌گونه ارزیابی‌ها است. امروزه سامانه‌های رایانه‌ای در بسیاری از کاربردهای حساس و بحرانی انسان نقش دارند به طوری‌که یک خطای کوچک، می‌تواند منجر به بروز مشکلات جدی و خطرناکی شود به همین علت آزمون نرم‌افزار به عنوان یک قسمت مهم در فرآیند تضمین کیفیت نرم‌افزار تلقی می‌شود. شرکت‌های نرم‌افزاری در مرحله‌ی آزمون نرم‌افزار برای حذف خطاهای برنامه تلاش زیادی داشته‌اند. در تولید نرم‌افزار، تخمین زده شده است که حدود ۶۰ درصد از فرآیند تولید نرم‌افزار صرف آزمون نرم‌افزار می‌شود. با این وجود شرکت‌ها نمی‌توانند با قاطعیت ادعا کنند که نرم‌افزارهای تولیدی عاری از خطا هستند. از طرفی با در نظر گرفتن زمان‌بندی پروژه از ابتدای شروع آن، مسئله‌ی تحویل به موقع محصول به مشتری در نظر گرفته می‌شود، اما برخی اوقات ممکن است به خاطر دیرکرد ساخت پروژه یا درخواست زود هنگام مشتری برای دریافت محصول، مدت زمان کافی برای آزمون و اشکال‌زدایی کل نرم‌افزار وجود نداشته باشد. در نتیجه مسئله‌ی کاهش زمان در فرآیند آزمون نرم‌افزار عامل مهمی محسوب می‌شود. در این فصل تعدادی از اصطلاحات، بیان مسئله و اهداف پایان‌نامه آورده شده است.

[1] software testing

۱-۱ اصطلاحات

آزمون نرم‌افزار: آزمون نرم‌افزار به دنبال خطایابی و عیب‌یابی محصول، قبل از تحویل به مشتری است [۱].

نمونه آزمون[1]: به نوشتن رویه‌هایی[2] در برنامه‌نویسی برای آزمودن قسمت‌های نرم‌افزار به صورت جزء به جزء، نمونه آزمون گفته می‌شود. یک نمونه آزمون با دادن متغیرها و به وجود آوردن شرایط واقعی برای رویه‌های عملیاتی نرم‌افزار باعث تحت آزمون قرار گرفتن رویه‌ها می‌شود و قابل اتکا بودن کد نرم‌افزار را می‌آزماید [۱].

خوشه‌بندی[3]: خوشه‌بندی به فرآیند گروه‌بندی اشیاء مشابه یکدیگر با هم، گفته می‌شود. به هر یک از گروه‌ها خوشه گفته می‌شود [۲].

اولویت‌بندی[4]: منظور از اولویت‌بندی، دادن امتیاز یا برتری به یک عنصر نسبت به بقیه عناصر است.

خطا[5]: نوعی اشتباه در اجرای نرم‌افزار است که موجب نتایج اشتباه یا اجرا نشدن نرم‌افزار می‌شود.

اتصال درون خوشه[6]: اتصال درون خوشه‌ها، تراکم اتصال‌های بین گره‌های داخل خوشه را اندازه می‌گیرد و بیشترین میزان اتصال‌های داخل خوشه بیانگر خوشه‌بندی بهتر است چون گره‌های داخل یک خوشه بیشترین وابستگی را به هم دیگر دارند [۳].

اتصال برون خوشه[7]: اتصال برون خوشه، اتصال‌های بین خوشه‌های مجزا را اندازه می‌گیرد که اگر این مقدار زیاد باشد مطلوب نیست. زیرا نشان‌دهنده‌ی وابستگی زیاد خوشه‌ها است [۳].

---

[1] test case
[2] method
[3] clustering
[4] priortizing
[5] error
[6] cohesion
[7] coupling

۱-۲ بیان مسئله

یکی از مسائل مهم در زمینه ساخت، تحویل و توسعه‌ی نرم‌افزار، تحویل نرم‌افزار کامل، بدون عیب و به موقع به مشتری است. برای کسب این اطمینان نیازمند آزمون نرم‌افزار هستیم. آزمون نرم‌افزار یکی از پرهزینه‌ترین و زمان‌برترین فعالیت در تولید نرم‌افزار است. روش‌های سنتی و دستی آزمون نرم‌افزار هزینه بالای زمانی و انسانی دارد، به این دلیل در سال‌های اخیر تلاش شده است که از راه‌کارهای خودکارسازی[1] آزمون نرم‌افزار با کمترین دخالت بشر در آن استفاده شود. زمینه‌های آزمون نرم‌افزار شامل موارد متعددی از جمله انتخاب نوع آزمون، انتخاب معیار آزمون، تولید نمونه آزمون و غیره است که همه‌ی این مراحل هزینه و زمان بسیاری را صرف می‌کند. از طرفی زمان یکی از عوامل مهم در آزمون نرم‌افزار است، چون برای آزمودن و تحویل نرم‌افزار زمان محدودی در اختیار است. پس برای آزمون نرم‌افزار باید همه این موارد لحاظ شود.

با توجه به اهمیت زمان در آزمون نرم‌افزار، راه‌کارهایی برای کاهش زمان ارائه شده است. تولید مجموعه آزمون مناسب و همچنین استفاده از خوشه‌بندی و اولویت‌بندی از جمله‌ی این راه‌کارها است. از طرفی معیارهایی هم برای تشخیص میزان خطاپذیری نرم‌افزار مطرح شده‌اند که در زمینه‌ی آزمون کمک بسیاری کرده است. هر یک از راه‌کارها و معیارهای مطرح شده دارای مزایا و معایبی است اما می‌توان به جای انتخاب یکی از این موارد، از ترکیب این راه‌کارها استفاده کرده و نتیجه‌ی بهتری از آزمون به دست آورد. از طرفی می‌توان علاوه بر تمرکز روی مسائل آزمون و روش‌های بهبود دادن آزمون، به خود نرم‌افزار نیز توجه نمود. در واقع با تحلیل نرم‌افزار، شناسایی مکان‌های مستعد خطا و غیره، به بهبود روند آزمون کمک کرد.

تمرکز این پایان‌نامه بر روی تحلیل نرم‌افزار جهت کاهش زمان آزمون است. از آنجایی‌که کد نرم‌افزار، بزرگ و پیچیده است نمی‌توان به راحتی آن را تجزیه و تحلیل کرد. به این دلیل برای درک بهتر نرم‌افزار آن را به صورت گراف وابستگی[2] نشان می‌دهند. اما برای نرم‌افزارهای بزرگ، این گراف نیز قابل درک نخواهد بود. یکی از

[1] automatic

[2] Module Dependency Graph

راه‌کارها برای این مسئله، خوشه‌بندی گراف وابستگی نرم‌افزار است. به‌وسیله‌ی خوشه‌بندی، گراف وابستگی به بخش‌های کوچکتر و قابل فهم‌تر تقسیم می‌شود. با استفاده از خوشه‌بندی در آزمون نرم‌افزار، می‌توان به جای آزمودن کل برنامه به صورت یکپارچه، خوشه‌ها را تک تک مورد آزمون قرار داد. با توجه به هدف کاهش زمان آزمون نرم‌افزار، این خوشه‌بندی، گره‌هایی از گراف که بیشترین اتصال درون خوشه‌ای را دارند در یک خوشه قرار می‌دهد. این کار باعث می‌شود وقتی در گرهی خطایی به وجود آمد و این خطا منتشر شد، مکان خطای اصلی و خطاهای منتشر شده در یک خوشه باشد. این پایان‌نامه برای یافتن بهترین خوشه‌بندی از الگوریتم رقابت استعماری[1] استفاده کرده است. در واقع پیدا کردن بهترین خوشه‌بندی را به عنوان مسئله‌ی بهینه‌سازی در نظر گرفته است. پس از اینکه خوشه‌بندی مناسبی از نرم‌افزار ارائه شد با توجه به اهداف مورد نظر می‌توان به این خوشه‌ها اولویت داد تا به ترتیب اولویت، بررسی شوند. برای تحقق هدف کاهش زمان آزمون نرم‌افزار، از معیارهایی جهت تشخیص میزان خطاپذیر بودن خوشه‌ها استفاده می‌شود و بر اساس این معیارها به خوشه‌ها اولویت داده می‌شود. در واقع خوشه‌ی دارای اولویت بالاتر، خوشه‌ی خطاپذیرتر شناخته می‌شود.

۱-۳ اهداف پایان‌نامه

هدف از این پایان‌نامه ارائه الگوریتم تکاملی برای خوشه‌بندی نرم‌افزار و اولویت‌بندی این خوشه‌ها در راستای کاهش زمان آزمون نرم‌افزار است. به عبارت دیگر هدف این پایان‌نامه ادغام اهداف زیر در مرحله‌ی آزمون نرم‌افزار است.

- به دست آوردن الگوریتم خوشه‌بندی مناسب جهت خوشه‌بندی نرم‌افزار
- به دست آوردن الگوریتم اولویت‌بندی مناسب برای خوشه‌ها با در نظر گرفتن میزان خطاپذیر بودن هر خوشه به منظور کمینه‌سازی زمان آزمون

[1] Imperialist Competitive Algorithm