Semnan University

Journal of Modeling in Engineering

Journal homepage: https://modelling.semnan.ac.ir/

ISSN: 2783-2538



Research Article

Design and Optimization of Constrained Covering Arrays in Combinatorial Testing Using Metaheuristic Algorithms

Azam Amraei [a], Sajad Esfandyari [b,*]

[a] *computer group, department of Kermanshah ACECR institute of higher education, Kermanshah, Iran*
[b] Department of Computer Engineering, Faculty of Engineering, Malayer University, Malayer, Iran

PAPER INFO

Paper history:

Received: 2022-03-13
Revised: 2022-05-12
Accepted: 2022-10-19

Keywords:
Constrained Covering Arrays;
Combinatorial Testing;
Evolutionary Optimization;
ROBDD Algorithm;
BBO Algorithm.

ABSTRACT

This paper addresses the analysis and solution of the problem of creating covering arrays (CA) in the context of combinatorial testing. Combinatorial testing, as a fundamental approach in software quality evaluation and testing, helps identify errors and weaknesses in software by combining various variables and conditions. The main goal of this study is to design and create a covering array capable of covering all possible subsets of variables and conditions. To achieve this, new and advanced metaheuristic algorithms are utilized, which have not been previously applied in combinatorial testing. These algorithms leverage evolution-based optimizations and algorithms inspired by natural mechanisms to solve the covering array problem. Specifically, the combination of the Biogeography-Based Optimization (BBO) algorithm with the ROBDD algorithm is introduced as an innovative approach to generate minimal test sequences in covering arrays. These metaheuristic algorithms are recognized for their high capability in solving optimization problems and finding optimal solutions. In this study, these methods are used to generate covering arrays that cover all combinatorial test conditions and provide the best possible results.

DOI: https://doi.org/10.22075/jme.2024.31282.2495

© 2024 Published by Semnan University Press.
This is an open access article under the CC-BY 4.0 license.(https://creativecommons.org/licenses/by/4.0/)

UNCORRECTED PROOF

*** Corresponding author.**
E-mail address: esfandyari@malayeru.ac.ir

How to cite this article:

مقاله پژوهشی

طراحی و بهینه‌سازی آرایه‌های پوشش با محدودیت در آزمون‌های ترکیبی با استفاده از الگوریتم‌های فراابتکاری

اعظم امرائی[۱] و سجاد اسفندیاری[۲،*]

اطلاعات مقاله

دریافت مقاله: ۱۳۹۵/۰۳/۲۰
بازنگری مقاله: ۱۳۹۵/۰۶/۲۰
پذیرش مقاله: ۱۳۹۵/۰۷/۱۵

واژگان کلیدی:
آرایه پوششی با محدودیت،
آزمون‌های ترکیبی،
بهینه‌سازی تکاملی،
الگوریتم ROBDD،
الگوریتم جغرافیای زیستی.

چکیده

در این مقاله، به تحلیل و حل مسأله ایجاد آرایه‌های پوششی در زمینه آزمون‌های ترکیبی پرداخته شده است. آزمون‌های ترکیبی به‌عنوان یک رویکرد اساسی در ارزیابی و تست کیفیت نرم‌افزار، از طریق ترکیب متغیرها و شرایط مختلف، به شناسایی خطاها و نقاط ضعف در نرم‌افزار کمک می‌کنند. هدف اصلی این پژوهش، طراحی و ایجاد آرایه پوششی است که قادر به پوشش تمام زیرمجموعه‌های ممکن از متغیرها و شرایط باشد. برای این منظور، از الگوریتم‌های فراابتکاری جدید و پیشرفته‌ای استفاده می‌شود که به ندرت در زمینه آزمون‌های ترکیبی به کار رفته‌است. این الگوریتم‌ها با بهره‌گیری از بهینه‌سازی‌های مبتنی بر تکامل و الگوریتم‌های الهام گرفته از مکانیسم‌های طبیعی، به حل مسأله ایجاد آرایه‌های پوشش پرداخته‌اند. به‌طور خاص، ترکیب الگوریتم جغرافیای زیستی (BBO) با الگوریتم ROBDD به‌منظور ایجاد دنباله‌های آزمون کمینه در آرایه‌های پوششی با محدودیت به‌عنوان رویکردی نوآورانه معرفی می‌شود. این الگوریتم‌های فراابتکاری به‌واسطه قدرت بالای خود در حل مسائل بهینه‌سازی و یافتن بهترین راه‌حل‌ها شناخته می‌شوند. در این پژوهش، از این روش‌ها برای تولید آرایه‌های پوششی استفاده شده است که تمام شرایط آزمون ترکیبی را پوشش داده و بهترین نتایج ممکن را ارائه می‌دهند.

DOI: https://doi.org/10.22075/jme.2024.31282.2495

© 2024 Published by Semnan University Press.
This is an open access article under the CC-BY 4.0 license.(https://creativecommons.org/licenses/by/4.0/)

UNCORRECTED PROOF

۱-مقدمه[۱]

در فرآیند آزمون نرم‌افزار، شناسایی و رفع خطاهای موجود در سیستم‌های پیچیده و بزرگ یکی از چالش‌های اصلی است. یکی از روش‌های مؤثر برای حل این مشکل، آزمون ترکیبی (Combinatorial Testing) است که به‌ویژه در سیستم‌های با تعداد زیادی پارامتر و حالات مختلف، کاربرد بسیاری دارد. آزمون ترکیبی با استفاده از ترکیب تعداد محدودی از ورودی‌ها، به‌دنبال شبیه‌سازی تعاملات مختلف میان پارامترهای سیستم است تا خطاهای احتمالی در اثر این تعاملات شناسایی شود [۱، ۲].

آزمون ترکیبی به‌عنوان یک رویکرد مؤثر در آزمون سیستم‌های پیچیده، با استفاده از ترکیب‌های مختلف از ورودی‌ها، قادر است به‌طور جامع شرایط مختلف را پوشش دهد. به‌طور مثال، فرض کنید یک سیستم دارای سه پارامتر است: پارامتر اول می‌تواند یکی از سه مقدار {A, B, C} را بپذیرد، پارامتر دوم می‌تواند یکی از دو مقدار {۱, ۲} را بپذیرد، و پارامتر سوم می‌تواند یکی از چهار مقدار {X, Y, Z, W} را بپذیرد. در حالت کلی، تعداد ترکیب‌های ممکن این سه پارامتر به صورت حاصل‌ضرب تعداد مقادیر هر پارامتر محاسبه می‌شود که در این مثال برابر با ۳ × ۲ × ۴ = ۲۴ ترکیب مختلف است [۳، ۴]. با این حال، این تعداد

* پست الکترونیک نویسنده مسئول: esfandyari@malayeru.ac.ir
۱. گروه کامپیوتر، موسسه عالی جهاد دانشگاهی کرمانشاه، کرمانشاه، ایران.
۲. دانشکده فنی مهندسی، دانشگاه ملایر، ملایر، ایران.

استناد به این مقاله: نحوه استناد فارسی در اینجا درج گردد.

ترکیب ممکن است بسیار زیاد باشد و نیاز به کاهش تعداد آزمون‌ها باشد بدون اینکه پوشش کاملی از تمامی شرایط و تعاملات سیستم از دست برود [۵].

برای حل این مشکل، از آرایه‌های پوشش (Covering Array) استفاده می‌شود. آرایه پوشش مجموعه‌ای از تست‌ها است که به‌گونه‌ای انتخاب می‌شود که تمام ترکیب‌های ممکن از پارامترها (در یک سطح مشخص از تعاملات) حداقل یک‌بار در آن حضور داشته باشد. به‌طور خاص، آرایه پوشش برای یک تعداد معین از پارامترها و مقادیر، تمام ترکیب‌های ممکن از N پارامتر را در ابعادی که نیاز است، پوشش می‌دهد[۶، ۷].

به‌عنوان مثال، اگر برای سه پارامتر {A, B, C} و مقادیر مربوط به هرکدام (همانطور که در بالا توضیح داده شد)، بخواهیم ترکیب‌های دو پارامتری (در سطح 2-way interaction) را در آزمون‌ها پوشش دهیم، آرایه پوشش باید به‌گونه‌ای طراحی شود که تمامی ترکیب‌های {A, B}، {A, C}، {B, C} را در خود داشته باشد. در این صورت، آرایه پوشش می‌تواند تعداد آزمون‌ها را کاهش دهد، در حالی که همچنان تمامی شرایط و تعاملات ممکن را پوشش می‌دهد [۸].

در برخی از موارد، علاوه بر نیاز به پوشش کامل تعاملات پارامترها، محدودیت‌هایی نیز برای تولید آرایه پوشش وجود دارد که آن را آرایه پوشش با محدودیت (Constrained Covering Array) می‌نامیم [۹]. این محدودیت‌ها می‌تواند شامل محدودیت در تعداد آزمون‌ها، محدودیت زمانی، یا حتی محدودیت‌های سیستم‌های واقعی باشد که می‌خواهیم برای آن‌ها آزمون طراحی کنیم. در چنین شرایطی، هدف این است که آرایه پوشش با تعداد کمینه آزمون‌ها، تمامی ترکیب‌های مورد نظر را با دقت و به‌طور مؤثر پوشش دهد، در حالی که از منابع سیستم به‌صورت بهینه استفاده شود [۱۰].

در این پژوهش، به‌منظور بهینه‌سازی فرآیند تولید آرایه‌های پوشش با محدودیت، از ترکیب الگوریتم جغرافیای زیستی-خفاش (BBO) و الگوریتم ROBDD (Reduced Ordered Binary Decision Diagram) استفاده شده است. هدف اصلی این است که با استفاده از این الگوریتم‌های پیشرفته و بهینه‌سازی آن‌ها، مجموعه آزمون کمینه با دقت و کارایی بالا تولید شود. در این روش، با بهره‌گیری از الگوریتم‌های فراابتکاری، علاوه بر کاهش تعداد آزمون‌ها، سرعت تولید آزمون‌ها نیز افزایش می‌یابد و از این طریق به پوشش کامل تعاملات سیستم دست می‌یابیم. این روش قادر است به‌طور مؤثر مشکلات بهینه‌سازی محلی را حل کند و در عین حال، از آزمون‌های تصادفی برای بهبود سرعت و دقت استفاده نماید.

در نهایت، این پژوهش قصد دارد یک رویکرد نوین در تولید آرایه‌های پوشش با محدودیت ارائه دهد که هم‌زمان با کاهش تعداد آزمون‌ها، تضمین کند که تمامی شرایط لازم برای آزمون سیستم به‌درستی پوشش داده شده است. این رویکرد می‌تواند به بهبود کیفیت و دقت آزمون‌های ترکیبی در سیستم‌های پیچیده و کاربردی کمک شایانی کند.

در جمع‌بندی مباحث این بخش، نوآوری‌های اصلی پژوهش حاضر را می‌توان به صورت زیر خلاصه کرد:

- ارائه یک چارچوب یکپارچه مبتنی بر گراف علت و معلول و نمودارهای ROBDD برای کاهش تعداد متغیرها و ساده‌سازی مدل در مسئله تولید آرایه‌های پوششی با محدودیت.
- طراحی یک الگوریتم ترکیبی BBO–ROBDD برای تولید دنباله‌های آزمون کمینه با پوشش صددرصدی تعاملات t-way در بازه ۲ تا ۶ در حضور قیود عملی.
- تعریف ساختار ماتریس پوشش و تابع برازندگی ویژه که وزن هر نمونه آزمون را بر اساس تعداد ترکیب‌های جدید پوشش‌داده‌شده محاسبه می‌کند و از این طریق فرایند جستجو را هدایت می‌کند.
- به‌کارگیری الگوریتم BBO به‌عنوان موتور بهینه‌سازی برای انتخاب مجموعه‌ای بهینه از نمونه‌های آزمون، به نحوی که هم‌زمان تعداد آزمون‌ها و تعداد متغیرهای مؤثر کاهش یافته و پوشش ۱۰۰ درصدی حفظ شود.
- انجام یک مطالعه موردی کامل روی سیستم خاموش‌سازی بویلر و مقایسه جامع با روش‌های مطرح در ادبیات از جمله Jenny، TConfig، HSS، CS، PSO، PSO-ROBDD، GS، Imp، PICT، TVG و نظایر آن. که نشان می‌دهد روش پیشنهادی در تمام سطوح تعامل بررسی‌شده، کمترین اندازه آرایه یا یکی از کوچک‌ترین اندازه‌ها را به دست می‌آورد.

این پژوهش در چند بخش سازماندهی شده است. بخش دوم به معرفی مفاهیم پایه و مبانی نظری آزمون ترکیبی و آرایه‌های پوشش می‌پردازد. در بخش سوم، الگوریتم‌های پیشنهادی برای تولید آرایه‌های پوشش با استفاده از ترکیب

UNCORRECTED PROOF

الگوریتم جغرافیای زیستی-خفاش و ROBDD شرح داده می‌شود. بخش چهارم به ارزیابی و تحلیل نتایج حاصل از الگوریتم‌های پیشنهادی می‌پردازد و در نهایت، بخش پنجم به نتیجه‌گیری و پیشنهادات برای پژوهش‌های آینده اختصاص دارد.

۲- آزمون ترکیبی (Combinatorial Testing)

آزمون ترکیبی یک روش جامع و مؤثر در فرآیند آزمون نرم‌افزار است که هدف آن شبیه‌سازی تعاملات مختلف میان پارامترهای سیستم به‌منظور شناسایی خطاها و مشکلات احتمالی در اثر این تعاملات است. در این نوع آزمون، به‌جای آزمودن تمام ترکیب‌های ممکن از ورودی‌ها، که می‌تواند تعداد بسیار زیادی ترکیب ایجاد کند، تنها ترکیب‌هایی انتخاب می‌شوند که تعاملات مهم و بحرانی میان پارامترها را پوشش دهند [۱۱].

آزمون ترکیبی به‌ویژه در سیستم‌هایی که تعداد زیادی پارامتر و مقدار دارند بسیار مفید است. به‌عنوان مثال، برای سیستم‌هایی با ۱۰ پارامتر و هر کدام با ۱۰ مقدار مختلف، تعداد ترکیب‌های ممکن به ۱۰ میلیارد می‌رسد، که به‌طور مستقیم انجام این تعداد آزمون غیرعملی است. در این حالت، با استفاده از آزمون ترکیبی می‌توانیم تنها با انتخاب یک زیرمجموعه کوچک از ترکیب‌های مختلف، تمام تعاملات مورد نیاز را پوشش دهیم و خطاهای اصلی را شناسایی کنیم [۱۲].

۲-۱- آرایه پوششی

آرایه پوششی یک مجموعه از آزمون‌ها است که به‌گونه‌ای انتخاب شده‌اند که تمام ترکیب‌های مورد نظر از پارامترها و مقادیر آن‌ها را پوشش دهند. در واقع، آرایه پوششی برای آزمون ترکیبی به‌کار می‌رود تا اطمینان حاصل شود که تمامی تعاملات بین پارامترها، حداقل یک‌بار در مجموعه آزمون‌ها حضور دارد [۱۳].

فرض کنید که یک سیستم ۳ پارامتر دارد. هر پارامتر می‌تواند از مجموعه‌ای از مقادیر مختلف انتخاب شود. هدف این است که تمامی ترکیب‌های مختلف از این پارامترها را در یک مجموعه کوچک از آزمون‌ها پوشش دهیم. به‌طور کلی، اگر سیستم N پارامتر با مقادیر مختلف داشته باشد، تعداد ترکیب‌های ممکن برابر با حاصل‌ضرب تعداد مقادیر هر پارامتر خواهد بود.

برای ایجاد یک آرایه پوششی، باید حداقل یک‌بار هر ترکیب از مقادیر پارامترها در یک یا چند آزمون ظاهر شود. این فرآیند معمولاً با استفاده از الگوریتم‌های بهینه‌سازی انجام می‌شود تا تعداد آزمون‌ها به حداقل برسد، در حالی که تمام ترکیب‌های ممکن از پارامترها پوشش داده شوند [۱۴].

مثال ۱: فرض کنید یک سیستم ۳ پارامتر دارد:

- پارامتر ۱ : {A, B}
- پارامتر ۲: {۱, ۲}
- پارامتر ۳: {X, Y}

در این حالت، تعداد ترکیب‌های ممکن برابر با ۲ × ۲ × ۲ = ۸ است. حالا می‌خواهیم آرایه‌ای ایجاد کنیم که تمامی این ترکیب‌ها را پوشش دهد. به‌طور معمول، این ترکیب‌ها در جدول ۱ نشان داده شده است.

جدول ۱- تعداد ترکیبات در آرایه پوشش

آزمون	پارامتر ۱	پارامتر ۲	پارامتر ۳
۱	A	۱	X
۲	A	۲	Y
۳	B	۱	Y
۴	B	۲	X

این آرایه پوششی ۴ آزمون تولید کرده است که تمام ۸ ترکیب ممکن از پارامترها را پوشش می‌دهد. که شرایط آن به شرح ذیل است:

- هر سطر در این جدول نشان‌دهنده یک آزمون است.
- تمام ترکیب‌های ممکن از مقادیر پارامترها در این جدول وجود دارد.
- تعداد آزمون‌ها به‌طور قابل توجهی کمتر از تعداد ترکیب‌های ممکن است، در حالی که همچنان تمام تعاملات پوشش داده شده‌اند.

۲-۲- آرایه پوششی با محدودیت

آرایه پوششی با محدودیت به نوعی از آرایه پوششی گفته می‌شود که علاوه بر پوشش تعاملات پارامترها، محدودیت‌هایی نیز برای تولید آزمون‌ها وجود دارد. این محدودیت‌ها می‌تواند شامل موارد زیر باشد[۱۵، ۱۶]:

- محدودیت در تعداد آزمون‌ها: ممکن است به‌دلیل محدودیت‌های زمانی یا منابع، نتوان تعداد زیادی آزمون را اجرا کرد.

UNCORRECTED PROOF

- محدودیت‌های سیستم: برای برخی از سیستم‌ها، محدودیت‌هایی در تولید و انجام آزمون‌ها وجود دارد که باید رعایت شود.
- محدودیت‌های مقیاس: در سیستم‌های بزرگ، ممکن است محدودیت‌های مقیاس باعث شود که برخی از ترکیب‌ها تنها با شرایط خاصی قابل ارزیابی باشند.

در این نوع آرایه‌ها، هدف این است که با اعمال محدودیت‌های موجود، همچنان بتوان تمامی تعاملات مورد نظر را پوشش داد و از بهینه‌ترین تعداد آزمون‌ها استفاده کرد.

مثال ۲: فرض کنید که در سیستم ۳ پارامتر، علاوه بر ترکیب‌های معمول، محدودیت‌هایی در تعداد آزمون‌ها وجود دارد. مثلاً تنها می‌توان ۳ آزمون را اجرا کرد. در این صورت، به‌جای ایجاد آرایه‌ای با ۴ آزمون (مانند مثال ۱)، باید تنها ۳ آزمون انتخاب کنیم که بتوانند تمامی ترکیب‌های ضروری را پوشش دهند (جدول ۲).

جدول ۲- تعداد ترکیبات در آرایه پوشش با محدودیت

آزمون	پارامتر ۱	پارامتر ۲	پارامتر ۳
۱	A	۱	X
۲	A	۲	Y
۳	B	۱	Y

در جدول ۲، به‌جای آرایه جدول ۱، ما تنها ۳ آزمون تولید کرده‌ایم که تمامی ترکیب‌های اصلی را پوشش می‌دهند. در این حالت، ممکن است برخی از ترکیب‌ها به‌طور غیرمستقیم در مجموعه آزمون‌ها پوشش داده شوند، ولی به دلیل محدودیت‌هایی که وجود دارد، تعداد آزمون‌ها کاهش یافته است.

آرایه پوششی با محدودیت‌ها می‌تواند به‌طور مؤثر در شرایطی که نیاز به کاهش تعداد آزمون‌ها وجود دارد، استفاده شود. این روش به‌ویژه در مواردی که منابع محدود است یا زمان محدودی برای انجام آزمون‌ها در اختیار داریم، می‌تواند مفید واقع شود. همچنان که در مثال‌ها مشاهده شد، می‌توان با کاهش تعداد آزمون‌ها، همچنان پوشش مطلوبی از ترکیب‌های مختلف پارامترها به‌دست آورد.

۳- الگوریتم ROBDD

الگوریتم ROBDD (Reduced Ordered Binary Decision Diagram) یک ساختار داده‌ای برای نمایش توابع منطقی به شکل فشرده است. این الگوریتم از نمودارهای دودویی استفاده می‌کند که در آن گره‌ها نمایانگر متغیرهای منطقی هستند و یال‌ها نشان‌دهنده نتایج مختلف آن‌ها (صحت یا غلط). پس از تعیین ترتیب متغیرها، نمودار ساخته شده و سپس گره‌های تکراری حذف می‌شوند تا یک نمودار بهینه باقی بماند [۱۷].

۳-۱- ویژگی‌ها:

- ترتیب متغیرها: ترتیب انتخاب متغیرها در ساخت نمودار اهمیت دارد.
- کاهش نمودار: گره‌های تکراری حذف می‌شوند.
- فشرده‌سازی داده‌ها: این الگوریتم به کاهش حجم داده‌ها کمک می‌کند.

۳-۲- کاربردها:

- بهینه‌سازی توابع منطقی: در طراحی مدارهای دیجیتال و بهینه‌سازی توابع.
- آزمون ترکیبی: برای تولید نمونه‌های آزمون بهینه در سیستم‌های پیچیده.
- مدل‌سازی و شبیه‌سازی: تحلیل سیستم‌های پیچیده و شبیه‌سازی رفتار آن‌ها.

۳-۳- مزایا:

- کاهش حجم داده‌ها
- پوشش جامع شرایط مختلف
- بهینه‌سازی در طراحی و تحلیل سیستم‌ها

ROBDD به دلیل فشرده‌سازی و بهینه‌سازی در تحلیل توابع منطقی، کاربردهای زیادی در طراحی سیستم‌های پیچیده و آزمون‌های نرم‌افزاری دارد.

۴- الگوریتم BBO

الگوریتم فراابتکاری BBO (Biogeography-Based Optimization) یک الگوریتمی است که از نظریه بیوگرافی زیستی الهام گرفته است. این الگوریتم برای حل مسائل بهینه‌سازی در فضای جستجوهای پیچیده و چندبعدی طراحی شده است و بر اساس فرآیندهای طبیعی مهاجرت و پراکندگی گونه‌ها در زیستگاه‌های مختلف عمل می‌کند. در واقع، BBO از فرآیندهای بوم‌شناختی موجود در طبیعت برای جستجوی راه‌حل‌های بهینه استفاده

UNCORRECTED PROOF

می‌کند [۱۸].

در این الگوریتم، هر راه‌حل به یک "زیستگاه" مشابه است که دارای ویژگی‌هایی است که به پارامترهای مسئله بهینه‌سازی مرتبط هستند. ویژگی‌های این زیستگاه‌ها به طور طبیعی توسط یک فرآیند مهاجرت تغییر می‌کنند. الگوریتم شامل دو مرحله اصلی است:

- مهاجرت: گونه‌ها (حل‌ها) از یک زیستگاه به زیستگاه دیگر مهاجرت می‌کنند و ویژگی‌های خود را با سایر گونه‌ها به اشتراک می‌گذارند. این مهاجرت‌ها موجب بهبود ویژگی‌های راه‌حل‌ها می‌شود.
- تبادل اطلاعات: در هر مرحله، ویژگی‌های گونه‌ها با یکدیگر مبادله می‌شود و زیستگاه‌ها به طور پویا تغییر می‌کنند تا به سمت راه‌حل‌های بهینه حرکت کنند.

از آنجایی که فرآیند مهاجرت موجب گسترش تنوع در فضای جستجو می‌شود، الگوریتم BBO می‌تواند به طور مؤثری در مسائل بهینه‌سازی چندهدفه و پیچیده به کار رود. به همین دلیل، این الگوریتم در کاربردهای مختلفی مانند بهینه‌سازی پارامترها، طراحی سیستم‌ها، و مسائل پیچیده دیگر کاربرد دارد. BBO به ویژه زمانی مؤثر است که فضای جستجو دارای چندین ماکزیمم محلی باشد و الگوریتم باید از این ماکزیمم‌ها عبور کرده و به بهینه‌سازی سراسری برسد.

۵- کارهای پیشین

تولید آرایه‌های پوششی یکی از مسائل مهم در آزمایش‌های ترکیبی است که به منظور شناسایی خطاهای ناشی از تعاملات بین اجزاء سیستم مورد استفاده قرار می‌گیرد. این فرآیند به دو دسته اصلی تقسیم می‌شود[۱۹، ۲۰]:

- روش‌های ریاضی: این روش‌ها بر پایه ساختارهای ترکیبی مانند آرایه‌های متعامد (Orthogonal Arrays) بنا شده‌اند و برای تنظیمات خاص مؤثر هستند.
- روش‌های محاسباتی [۲۱]: این روش‌ها از تکنیک‌های جستجوی فرامکاشفه ای و هوش مصنوعی برای تولید آرایه‌های پوششی استفاده می‌کنند و می‌توانند با موقعیت‌های گسترده‌تری برخورد کنند.

در روش‌های محاسباتی، دو رویکرد اصلی برای تولید موارد آزمون وجود دارد [۲۲]:

- یک تست در هر آزمون (OTAT): در این روش، هر مورد آزمون تنها یک ترکیب خاص از مقادیر ورودی را پوشش می‌دهد. راهکارهای برجسته مانند AETG، mAETG، PICT، DDA، CTEX-XL، TVG، Jenny، GTWay و ITC-H را می‌توان در این دسته قرار داد.
- یک پارامتر در هر آزمون (OPAT): این روش با تغییر یک پارامتر در هر بار و نگه داشتن سایر پارامترها ثابت، ترکیب‌های مختلف مقادیر ورودی را پوشش می‌دهد. راهکارهای معمولی مانند IPOG، IPOG-s و IPOG-D در این دسته قرار می‌گیرند.

با افزایش پیچیدگی سیستم‌ها، استفاده از الگوریتم‌های هوش مصنوعی برای تولید آرایه‌های پوششی مورد توجه قرار گرفته است. برخی از این الگوریتم‌ها عبارت‌اند از:

- شبیه‌سازی تبرید (SA): این الگوریتم با الهام از فرآیند خنک‌سازی فلزات، به بهینه‌سازی آرایه‌های پوششی می‌پردازد [۲۳].
- جستجوی تابو (TS): با استفاده از حافظه کوتاه‌مدت، به جستجوی فضای حل می‌پردازد تا از تکرار راه‌حل‌های قبلی جلوگیری کند [۲۴، ۲۵].
- الگوریتم ژنتیک (GA): با تقلید از فرآیندهای انتخاب طبیعی، به تولید آرایه‌های پوششی می‌پردازد [۹].
- بهینه‌سازی توده ذرات (PSO): با الهام از رفتار گروهی پرندگان، به جستجوی فضای حل می‌پردازد [۲۶].
- الگوریتم کوکو (CS): با تقلید از رفتار زنبورها، به بهینه‌سازی آرایه‌های پوششی می‌پردازد [۱۳].
- بهینه‌سازی هارمونی (HS): با الهام از فرآیندهای موسیقی، به تولید آرایه‌های پوششی می‌پردازد [۲۷].
- الگوریتم چند چاله سیاه (MBH): با استفاده از اصول چاله‌های سیاه، به تولید آرایه‌های پوششی می‌پردازد.
- الگوریتم بهینه‌سازی تولید جنسیت‌نامعین (ARO): با استفاده از تکنیک‌های فازی و ژنتیکی، به بهینه‌سازی آرایه‌های پوششی می‌پردازد [۱].
- الگوریتم ممتیک QSSMA: با ترکیب الگوریتم‌های ژنتیک و جستجوی محلی، به تولید آرایه‌های پوششی با قدرت بالا می‌پردازد [۲۸].

در سال ۲۰۰۱، Stardom با استفاده از الگوریتم‌های SA، GA و TS، مجموعه آزمونی با تعامل قدرت ۲ ایجاد کرد

UNCORRECTED PROOF

که الگوریتم SA بهترین عملکرد را نشان داد. همچنین، الگوریتم‌های PSO و مشتقات آن مانند DPSO و BAPSO توانایی پشتیبانی از تعاملات با قدرت‌های بالاتر را دارند، اما ممکن است با مشکلاتی مانند همگرایی زودهنگام مواجه شوند. الگوریتم GTWay نیز به دلیل استفاده از ساختار بیتی برای ذخیره موارد آزمون، قادر به تولید موارد آزمون با تعاملات بالاتر است.

با توجه به پیشرفت‌های اخیر، الگوریتم‌های جدیدی مانند ImpARO و QSSMA توسعه یافته‌اند که با ترکیب تکنیک‌های مختلف، به بهبود کارایی و پوشش تعاملات پیچیده‌تر کمک می‌کنند.

در پژوهش دیگر [۲۹] مسئله تولید آرایه‌های پوششی با محدودیت در چارچوب آزمون‌های ترکیبی بررسی شده و با ترکیب گراف‌های ROBDD و الگوریتم فرااابتکاری BBO، روشی برای تولید مجموعه آزمون کمینه با پوشش کامل تعاملات ارائه شده است. نتایج عددی نشان می‌دهد رویکرد پیشنهادی در مقایسه با روش‌های مرجع، تعداد تست‌ها را به‌طور معنی‌داری کاهش می‌دهد، در حالی که پوشش صددرصدی ترکیب‌های پارامترها حفظ می‌شود.

۶- پیاده سازی راهکار پیشنهادی

در سیستم‌های امروزی، قابلیت اعتماد نرم‌افزارها نقش مهمی در دستیابی به اهداف آن‌ها دارد. برای اطمینان از صحت عملکرد سیستم، آزمون نرم‌افزار با مجموعه‌ای از نمونه‌های آزمون ضروری است. با این حال، اجرای تمامی نمونه‌های آزمون ممکن است زمان‌بر باشد. بنابراین، انتخاب مجموعه‌ای بهینه از دنباله‌های آزمون که پوشش کامل را فراهم کند، اهمیت دارد. در این راستا، روش پیشنهادی با بهره‌گیری از آرایه پوششی و ترکیب الگوریتم‌های بهینه‌سازی، به تولید دنباله‌های آزمون کمینه و مؤثر می‌پردازد. در این بخش، مراحل اجرای الگوریتم پیشنهادی به تفصیل ارائه خواهد شد.

۶-۱- تولید جمعیت اولیه

گام نخست در راهکار پیشنهادی، تولید جمعیت اولیه بر اساس الگوریتم بهینه‌سازی جغرافیایی زیستی (BBO) است. این روش به منظور تولید دنباله‌های آزمون مبتنی بر آرایه پوششی، هم در شرایط دارای محدودیت (با استفاده از الگوریتم ROBDD) و هم بدون محدودیت، به کار گرفته می‌شود. هدف اصلی، کاهش تعداد نمونه‌های آزمون در فرآیند تست است. در این مرحله، پارامترهای اولیه الگوریتم BBO مقداردهی می‌شوند. در این الگوریتم، هر فرد معادل یک نمونه آزمون است که شامل تعدادی پارامتر با مقادیر مشخص در یک پیکربندی معین می‌باشد. به عنوان مثال، در پیکربندی CA (N; 2, 6, 3)، جمعیت اولیه شامل نمونه‌هایی است که هر پارامتر آن دارای یکی از سه مقدار ممکن است.

۶-۲- تولید ماتریس پوشش

در مرحله بعد، محاسبه وزن نمونه‌های آزمون نقش کلیدی در فرایند تولید دنباله آزمون دارد. با اضافه شدن هر نمونه آزمون، تعداد پوشش‌های جدیدی که تاکنون لحاظ نشده‌اند، باید تعیین شود. برای این منظور، از یک ساختار داده‌ای با دسترسی سریع به داده‌ها استفاده شده است. در این بخش، نحوه ایجاد ماتریس پوشش و محاسبه وزن نمونه آزمون بررسی می‌شود. تابع برازندگی مسئول تعیین وزن هر نمونه آزمون است، به‌طوری‌که وزن بالاتر نشان‌دهنده اهمیت بیشتر آن در پوشش مؤثر ترکیبات است. زمانی که مجموع وزن در ساختار $CA(N;t,p,v)$ به پوشش کامل برسد، دنباله آزمون به پوشش کامل دست یافته و الگوریتم پایان می‌یابد. در ماتریس پوشش، تعداد سطرها برابر با $\binom{p}{t}$ و تعداد ستون‌ها برابر با $v^t + t$ است. به‌عنوان نمونه، شکل ۱ (الف) ماتریس پوشش برای ساختار $CA(N;2,4,2)$ را نشان می‌دهد که در آن، مقدار ۱ و ۲ در سطر نخست، ترکیب ستون‌های اول و دوم را مشخص می‌کنند، درحالی‌که سایر ستون‌ها مقدار صفر دارند که به معنای عدم پوشش است.

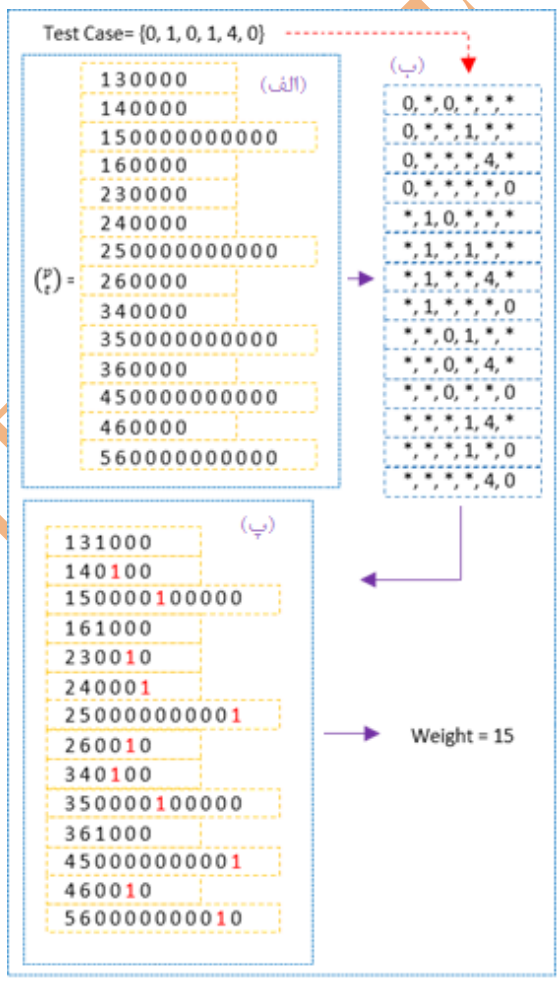
۶-۳- محاسبه وزن یک نمونه آزمون

در راهکار پیشنهادی، ابتدا مقدار دهدهی هر نمونه آزمون محاسبه می‌شود. اگر مقدار متناظر در ماتریس پوشش برابر صفر باشد، مقدار آن به یک تغییر کرده و وزن نمونه آزمون افزایش می‌یابد. در مثال ارائه‌شده، ۶ ترکیب مختلف وجود دارد که هرکدام شامل ۴ مقدار ممکن (۰۰، ۰۱، ۱۰، ۱۱) بوده و در مجموع ۲۴ حالت پوشش‌نیافته اولیه را تشکیل می‌دهند.

به عنوان نمونه، مقدار ۰۱۰۱ در نظر گرفته می‌شود. در گام نخست، ترکیب پارامترهای ۱ و ۲ انتخاب شده و مقدار

UNCORRECTED PROOF

معادل آن ۰۱ است (شکل ۱ (ب)). این مقدار به یک موقعیت خاص در ماتریس پوشش مرتبط است. اگر مقدار این موقعیت صفر باشد، مقدار آن به یک تغییر یافته و وزن نمونه آزمون افزایش می‌یابد (شکل ۱ (پ)). این روند برای ترکیب‌های دیگر (مانند ترکیب ۱ و ۳ با مقدار ۰۰) تکرار می‌شود تا تمامی ترکیب‌های موردنظر بررسی شوند. در نهایت، وزن محاسبه‌شده این نمونه آزمون ۱۵ واحد خواهد بود.



شکل ۱- ساختار ذخیره سازی نمونه آزمون

انتخاب نمونه آزمون بر عهده الگوریتم BBO است. هر نمونه آزمون انتخاب‌شده، وزن خود را از مجموعه حالت‌های پوشش‌نیافته کم کرده و این فرآیند تا زمانی ادامه می‌یابد که تمامی مقادیر صفر در ماتریس پوشش به یک تبدیل شوند، به این ترتیب، پوشش کامل حاصل خواهد شد.

۶-۴- خلاصه سازی با الگوریتم ROBDD

در این مطالعه، رویکردی نوین برای بهینه‌سازی و ساده‌سازی عبارات منطقی در تحلیل گراف علت و معلول ارائه شده است. این روش با ترکیب گراف‌های ROBDD و الگوریتم BBO، ضمن کاهش تعداد گره‌ها، موجب بهبود کارایی و کاهش زمان پیمایش می‌شود. ازآنجاکه ROBDD نسخه‌ای بهینه و منحصربه‌فرد از هر عبارت منطقی ارائه می‌دهد، این رویکرد علل غیرضروری را حذف کرده و هزینه و زمان اجرای آزمون‌ها را کاهش می‌دهد. افزون بر این، الگوریتم BBO برای یافتن مجموعه‌ای بهینه از نمونه‌های آزمون با پوشش کامل به کار گرفته شده‌اند.

مراحل اجرایی این رویکرد عبارت‌اند از:

- طراحی گراف علت و معلول برای هر ویژگی،
- تعریف ویژگی‌های سیستم در فرم نرمال INF،
- استخراج عبارات منطقی از گراف،
- ترسیم درخت تصمیم باینری،
- تبدیل درخت تصمیم به گراف ROBDD بهینه‌شده،
- در نظر گرفتن هر پیمایش در گراف ROBDD به عنوان یک نمونه آزمون.

۶-۵- تولید آرایه پوشش با الگوریتم BBO

الگوریتم پیشنهادی برای ساخت $CA(N: t, p, v)$ شامل دو مؤلفه اصلی برای هر عضو است: داده‌ها و وزن. داده‌ها در قالب آرایه‌ای با طول p ذخیره شده و مقادیر آن بین ۰ تا $v-1$ متغیر هستند. وزن هر نمونه آزمون از طریق تابع برازندگی محاسبه می‌شود. در آغاز الگوریتم، وزن اولیه تمامی نمونه‌ها برابر با ترکیب t از p در نظر گرفته می‌شود. نخستین نمونه به‌صورت تصادفی انتخاب شده که این امر به کاهش زمان تولید کمک می‌کند.

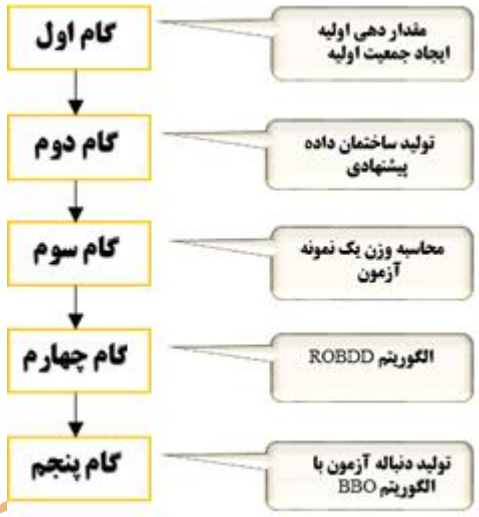
این الگوریتم دارای دو حلقه اصلی است:

- حلقه اول: تعیین تعداد کل نمونه‌های آزمون، که در هر دور یک نمونه جدید به مجموعه آزمون افزوده می‌شود. الگوریتم BBO مسئول انتخاب نمونه‌های بعدی است، درحالی‌که در هر مرحله، بهترین مقدار اولیه به‌طور تصادفی انتخاب می‌شود.
- حلقه دوم: تکرار برای جمعیت، که بسته به پارامترهای ورودی CA بین ۲۰ تا ۵۰ بار اجرا می‌شود. در این حلقه، از عملیات کراس‌اوور و جهش برای بهبود نمونه‌ها استفاده می‌شود. در BBO، جمعیت‌های باارزش کمتر دچار تغییرات بیشتری شده، درحالی‌که نمونه‌های

UNCORRECTED

باارزش کمتر اصلاح می‌شوند. تابع جهش نیز با اعمال تغییرات تصادفی به بهینه‌سازی کمک می‌کند.

فلوچارت ارائه‌شده در شکل ۲، مراحل اجرای این الگوریتم را به‌صورت گام‌به‌گام نمایش می‌دهد.



شکل ۲- مراحل راهکار پیشنهادی

۷- ارزیابی و مقایسه راهکارها

در این پژوهش، ارزیابی بر اساس معیار بهره‌وری (اندازه آرایه) انجام شده است. بهره‌وری تنها به فرآیند اجرای راهکار بستگی دارد، در حالی که کارایی مستقیماً تحت تأثیر زیرساخت‌های سخت‌افزاری و نرم‌افزاری قرار دارد. با توجه به تنوع روش‌های تولید آرایه‌های پوششی، امکان بررسی تمامی آن‌ها وجود ندارد. ازاین‌رو، برای مقایسه‌ای عادلانه، تعدادی از روش‌های موجود انتخاب شده‌اند.

از میان روش‌های مبتنی بر هوش مصنوعی، CPSO، DPSO و GS انتخاب شده‌اند که در این میان، DPSO از نظر بهره‌وری بهترین عملکرد را ارائه داده است. البته ممکن است برخی روش‌های دیگر نتایج بهتری داشته باشند، اما به دلیل محدودیت منابع، این پژوهش بر این سه روش متمرکز شده است.

همچنین، برای مقایسه با روش‌های غیرهوش مصنوعی، Jenny، TConfig، PICT و IPOG بررسی شده‌اند که از الگوریتم‌های قطعی محسوب می‌شوند.

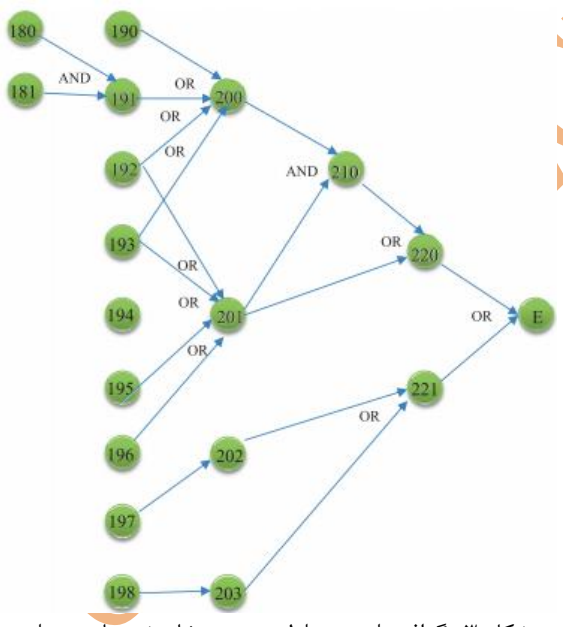
۷-۱- مشخصات زیرساخت و نحوه اجرا

- سیستم‌عامل: Windows 7
- پردازنده: Intel Core™ i7QM 2.20GHz
- حافظه: RAM 6GB
- محیط برنامه‌نویسی: Eclipse با JDK 1.8

روش‌های غیرهوش مصنوعی نتایج قطعی تولید می‌کنند، درحالی‌که روش‌های هوش مصنوعی غیرقطعی بوده و ممکن است در هر اجرا نتایج متفاوتی ارائه دهند. برای کاهش اثرات این عدم قطعیت، هر پیکربندی ۱۰۰ بار اجرا شده و دو معیار بهترین (Best) و میانگین (Average) نتایج گزارش شده است. اطلاعات مربوط به روش‌های GS، CPSO و DPSO در صورت دسترسی، مستقیماً از مقالات استخراج شده‌اند؛ در غیر این صورت، تعداد اجراها مشابه الگوریتم پیشنهادی تنظیم شده است.

۷-۲- فرآیند پیشنهادی

پژوهش حاضر روشی جدید ارائه می‌دهد که در آن، ابتدا نیازمندی‌های نرم‌افزاری از کاربر در قالب فرمت INF دریافت شده و سپس یک گراف علت و معلول مطابق با این نیازمندی‌ها تهیه می‌گردد. نمونه‌ای از این گراف در شکل ۳ نمایش داده شده که علت و معلول‌های مرتبط با سیستم خاموش‌کننده بویلر را نشان می‌دهد.



شکل ۳- گراف علت و معلول سیستم خاموش سازی بویلر

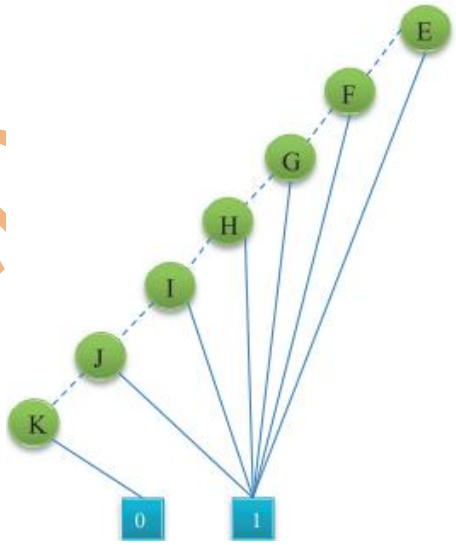
پس از تهیه گراف علت و معلول و استخراج عبارت منطقی مرتبط، گام بعدی تولید گراف ROBDD است. با استفاده از این روش، تعداد متغیرهای سیستم از ۱۰ به ۷ کاهش یافته است (شکل ۴). این کاهش، تأثیر چشمگیری بر کاهش زمان اجرای آزمون‌ها و هزینه‌های تست دارد، زیرا:

- تعداد مسیرهای محاسباتی کاهش می‌یابد.

- پیچیدگی پردازش داده‌ها کم‌تر می‌شود.
- تعداد نمونه‌های آزمون موردنیاز کاهش می‌یابد.

با پیمایش مسیرهای گراف ROBDD، مجموعه‌ای از نمونه‌های آزمون تولید می‌شود. این فرآیند باعث می‌شود که تنها ترکیب‌های ضروری در نظر گرفته شوند و نمونه‌های آزمون غیرضروری حذف گردند. درنتیجه، مجموعه آزمون حاصل، مناسب‌ترین پوشش ممکن را برای سیستم ارائه می‌دهد.

برای انتخاب مجموعه‌ای بهینه از تست‌ها، از الگوریتم بهینه‌سازی زیست‌جغرافیایی (BBO) استفاده شده است. پارامترهای تنظیم‌شده برای این الگوریتم در جدول ۳ ارائه شده‌اند.



شکل ۴- گراف ROBDD سیستم خاموش سازی بویلر

جدول ۳- پارامترهای الگوریتم پیشنهادی برای سیستم خاموش سازی بویلر

نام پارامتر	نام متغیر	مقدار
تعداد گره‌های گراف ROBDD	n	۷
قوه تعامل	t	۲ تا ۶
حداکثر تعداد تکرار	iteration	۱۰۰۰
تعداد مقادیر پارامترها	v	۲

۷-۳- ارزیابی

نتایج ارزیابی روش پیشنهادی نشان می‌دهد که در مقایسه با سایر روش‌ها، کمترین تعداد متغیرها و نمونه‌های آزمون را داراست. این کاهش، که به کمک گراف‌های ROBDD و الگوریتم BBO حاصل شده است، تأثیر بسزایی در افزایش سرعت تست و کاهش هزینه‌های آن داشته، در حالی که پوشش ۱۰۰ درصدی تست همچنان حفظ شده است. کاهش تعداد متغیرها باعث ساده‌سازی فرآیند تست و کاهش پیچیدگی پردازشی شده و کاهش تعداد نمونه‌های آزمون نیز منجر به بهینه‌سازی منابع محاسباتی و کاهش هزینه‌های اجرایی گردیده است. بنابراین، روش پیشنهادی نه‌تنها از نظر دقت برتری دارد، بلکه به دلیل کاهش نیاز به منابع پردازشی، یک راهکار مقرون‌به‌صرفه و کارآمد محسوب می‌شود.

جدول ۴ نتایج مقایسه‌ای بین روش پیشنهادی و سایر روش‌های موجود را برای قوه تعامل ۲ نشان می‌دهد. این مقایسه شامل تعداد نمونه‌های آزمون، میزان پوشش آزمون، و تعداد متغیرهای استفاده‌شده در هر روش است. همان‌طور که مشاهده می‌شود، روش پیشنهادی و روش PSO-ROBDD با کمترین تعداد نمونه آزمون (۶ مورد) و حفظ پوشش ۱۰۰ درصدی، بهترین عملکرد را ارائه می‌دهند. همچنین، این دو روش توانسته‌اند تعداد متغیرها را از ۱۰ به ۷ کاهش دهند، که نشان‌دهنده بهینه‌سازی ساختاری و کاهش پیچیدگی تست است. سایر روش‌های مبتنی بر الگوریتم‌های کلاسیک و هوش مصنوعی مانند Jenny، TConfig، HSS، CS، PSO، GS، Imp، PICT و TVG همگی تعداد بیشتری از نمونه‌های آزمون را نیاز دارند (بین ۷ تا ۱۰ نمونه) و همگی دارای ۱۰ متغیر هستند. این نتایج نشان می‌دهد که روش پیشنهادی، علاوه بر حفظ دقت تست، موجب کاهش تعداد متغیرها و نمونه‌های آزمون شده است که این امر در کاهش هزینه‌ها و افزایش سرعت تست تأثیر بسزایی دارد.

جدول ۴- مقایسه روش‌های مختلف برای قوه تعامل ۲

روش	تعداد نمونه آزمون	میزان پوشش	تعداد متغیر
روش پیشنهادی	۶	۱۰۰٪	۷
PSO-ROBDD	۶	۱۰۰٪	۷
Jenny	۱۰	۱۰۰٪	۱۰
TConfig	۹	۱۰۰٪	۱۰
HSS	۷	۱۰۰٪	۱۰
CS	۷	۱۰۰٪	۱۰
PSO	۸	۱۰۰٪	۱۰
GS	۷	۱۰۰٪	۱۰
Imp	۷	۱۰۰٪	۱۰
PICT	۹	۱۰۰٪	۱۰
TVG	۹	۱۰۰٪	۱۰

UNCORRECTED PROOF

جدول ۵ نشان‌دهنده مقایسه روش پیشنهادی با سایر روش‌های موجود برای قوه تعامل ۳ است. روش پیشنهادی توانسته با کمترین تعداد نمونه آزمون (۱۲ مورد) و حفظ پوشش ۱۰۰ درصدی، بهترین عملکرد را ارائه دهد، درحالی‌که سایر روش‌ها به تعداد نمونه‌های بیشتری نیاز داشته‌اند. همچنین در این مقایسه نیز، روش پیشنهادی و روش PSO-ROBDD تعداد متغیرها را از ۱۰ به ۷ کاهش داده‌اند، که منجر به کاهش پیچیدگی و افزایش بهره‌وری فرآیند تست می‌شود. سایر روش‌ها، هرچند میزان پوشش ۱۰۰ درصدی را حفظ کرده‌اند، اما به تعداد بیشتری از نمونه‌های آزمون نیاز دارند، که می‌تواند موجب افزایش هزینه و زمان اجرای آزمون‌ها شود.

جدول ۵- مقایسه روش‌های مختلف برای قوه تعامل ۳

روش	تعداد نمونه آزمون	میزان پوشش	تعداد متغیر
روش پیشنهادی	۱۲	۱۰۰٪	۷
PSO-ROBDD	۱۵	۱۰۰٪	۷
Jenny	۱۹	۱۰۰٪	۱۰
TConfig	۲۰	۱۰۰٪	۱۰
HSS	۱۶	۱۰۰٪	۱۰
CS	۱۶	۱۰۰٪	۱۰
PSO	۱۶	۱۰۰٪	۱۰
GS	۱۶	۱۰۰٪	۱۰
Imp	۱۵	۱۰۰٪	۱۰
PICT	۲۰	۱۰۰٪	۱۰
TVG	۲۰	۱۰۰٪	۱۰

در جدول ۶ که مربوط به قوه تعامل ۴ است، می‌توان مشاهده کرد که روش پیشنهادی با ۲۴ نمونه آزمون و ۷ متغیر، کمترین تعداد نمونه آزمون و متغیر را در مقایسه با سایر روش‌ها دارد. این در حالی است که سایر روش‌ها همچنان پوشش ۱۰۰٪ را ارائه می‌دهند، اما تعداد نمونه آزمون و متغیرهای مورد نیاز آن‌ها بیشتر است. به‌طور خاص، روش‌های مانند PICT و TVG که در این قوه تعامل به ترتیب ۴۵ و ۴۶ نمونه آزمون نیاز دارند، بیشترین تعداد نمونه آزمون را دارند، در حالی که روش‌های مبتنی بر هوش مصنوعی همچون PSO-ROBDD و PSO تعداد کمتری از نمونه‌های آزمون را می‌طلبند، اما همچنان با تعداد متغیرهای بالا روبه‌رو هستند.

جدول ۶- مقایسه روش‌های مختلف برای قوه تعامل ۴

روش	تعداد نمونه آزمون	میزان پوشش	تعداد متغیر
روش پیشنهادی	۲۴	۱۰۰٪	۷
PSO-ROBDD	۲۵	۱۰۰٪	۷
Jenny	۴۵	۱۰۰٪	۱۰
TConfig	۴۳	۱۰۰٪	۱۰
HSS	۲۶	۱۰۰٪	۱۰
CS	۲۵	۱۰۰٪	۱۰
PSO	۲۶	۱۰۰٪	۱۰
GS	۲۵	۱۰۰٪	۱۰
Imp	۳۳	۱۰۰٪	۱۰
PICT	۴۶	۱۰۰٪	۱۰
TVG	۴۵	۱۰۰٪	۱۰

در جدول ۷ که مربوط به قوه تعامل ۵ است، مشاهده می‌شود که روش پیشنهادی با ۴۸ نمونه آزمون و ۷ متغیر همچنان کمترین تعداد نمونه آزمون را در میان روش‌های مختلف دارد. این در حالی است که دیگر روش‌ها همچون PSO-ROBDD با ۵۳ نمونه آزمون و متغیر مشابه، یا روش‌هایی مانند Jenny و TConfig با ۸۷ و ۹۵ نمونه آزمون به ترتیب، نیاز به تعداد بیشتری از نمونه‌های آزمون دارند. این جدول نشان می‌دهد که با افزایش قوه تعامل، تعداد نمونه‌های آزمون در بسیاری از روش‌ها افزایش یافته، اما روش پیشنهادی هنوز هم به‌طور قابل توجهی از نظر بهینه بودن در تعداد نمونه آزمون نسبت به سایر روش‌ها برتری دارد.

جدول ۷- مقایسه روش‌های مختلف برای قوه تعامل ۵

روش	تعداد نمونه آزمون	میزان پوشش	تعداد متغیر
روش پیشنهادی	۴۸	۱۰۰٪	۷
PSO-ROBDD	۵۳	۱۰۰٪	۷
Jenny	۸۷	۱۰۰٪	۱۰
TConfig	۹۵	۱۰۰٪	۱۰
HSS	۸۰	۱۰۰٪	۱۰
CS	۷۹	۱۰۰٪	۱۰
PSO	۸۲	۱۰۰٪	۱۰
GS	۷۹	۱۰۰٪	۱۰
Imp	۷۹	۱۰۰٪	۱۰
PICT	۸۷	۱۰۰٪	۱۰
TVG	۸۶	۱۰۰٪	۱۰

UNCORRECTED PROOF

در جدول ۸ که مربوط به قوه تعامل ۶ است، روش پیشنهادی با ۶۴ نمونه آزمون و ۷ متغیر همچنان کمترین تعداد نمونه آزمون را در مقایسه با دیگر روش‌ها دارد. در حالی که سایر روش‌ها، مانند PSO-ROBDD با ۶۵ نمونه آزمون و روش‌هایی نظیر Jenny و TConfig با ۱۶۹ و ۱۸۳ نمونه آزمون، نیاز به تعداد بیشتری از نمونه‌های آزمون دارند. این جدول نشان می‌دهد که با افزایش قوه تعامل، تعداد نمونه‌های آزمون در اغلب روش‌ها افزایش یافته، اما روش پیشنهادی هنوز به‌طور قابل توجهی از نظر تعداد نمونه آزمون بهینه است. این ویژگی به کاهش هزینه و زمان اجرای آزمون‌ها کمک شایانی می‌کند.

جدول ۸- مقایسه روش‌های مختلف برای قوه تعامل ۶

روش	تعداد نمونه آزمون	میزان پوشش	تعداد متغیر
روش پیشنهادی	۶	۱۰۰٪	۷
PSO-ROBDD	۶	۱۰۰٪	۷
Jenny	۱۰	۱۰۰٪	۱۰
TConfig	۹	۱۰۰٪	۱۰
HSS	۷	۱۰۰٪	۱۰
CS	۷	۱۰۰٪	۱۰
PSO	۸	۱۰۰٪	۱۰
GS	۷	۱۰۰٪	۱۰
Imp	۷	۱۰۰٪	۱۰
PICT	۹	۱۰۰٪	۱۰
TVG	۹	۱۰۰٪	۱۰

۸- بحث و گفتگو

در این پژوهش، هدف اصلی طراحی و ارزیابی یک روش بهینه برای تولید مجموعه آزمون با پوشش ۱۰۰ درصد در سیستم‌های پیچیده با استفاده از الگوریتم‌های هوش مصنوعی و به ویژه الگوریتم BBO است. در مقایسه با روش‌های متداول مانند PSO-ROBDD و روش‌های غیر هوش مصنوعی همچون Jenny و TConfig، روش پیشنهادی به طور قابل توجهی از نظر تعداد نمونه‌های آزمون و تعداد متغیرها بهینه‌تر عمل کرده است.

در مراحل مختلف، با افزایش قوه تعامل از ۲ تا ۶، نتایج نشان می‌دهد که روش پیشنهادی توانسته است تعداد نمونه‌های آزمون را به طور محسوسی کاهش دهد، در حالی که پوشش ۱۰۰ درصدی را حفظ کرده است. به عنوان مثال، در قوه تعامل ۲، تعداد نمونه‌های آزمون روش پیشنهادی برابر با ۶ بود، در حالی که برای دیگر روش‌ها این مقدار بیشتر از ۷ بود. این روند به طور مشابه در قوه تعامل‌های بالاتر ادامه یافته است. در قوه تعامل ۶، که پیچیدگی سیستم به شدت افزایش می‌یابد، روش پیشنهادی همچنان با ۶۴ نمونه آزمون بهترین عملکرد را از خود نشان داده است.

همچنین، این پژوهش نشان می‌دهد که با استفاده از گراف‌های ROBDD و ترکیب آنها با الگوریتم BBO، بهینه‌سازی‌هایی در سطح گراف‌های علت و معلولی صورت گرفته که موجب کاهش چشمگیر تعداد متغیرها و در نتیجه کاهش زمان و هزینه آزمون‌ها شده است. این رویکرد، علاوه بر اینکه باعث بهبود عملکرد الگوریتم‌ها می‌شود، کمک شایانی به شناسایی علل غیرضروری و حذف آنها از فرآیند آزمون‌ها کرده است.

در نهایت، ارزیابی‌های انجام شده در این پژوهش، که شامل مقایسه روش‌های پیشنهادی با سایر روش‌های موجود است، نشان می‌دهند که روش‌های مبتنی بر هوش مصنوعی در شرایط خاص، به ویژه زمانی که پیچیدگی سیستم افزایش می‌یابد، می‌توانند گزینه‌های بهتری برای تولید مجموعه‌های آزمون بهینه و صرفه‌جویی در زمان و هزینه باشند.

۸-۱- بحث در خصوص کارهای اخیر

برای روشن‌تر شدن جایگاه روش پیشنهادی نسبت به کارهای اخیر، مقایسه‌ای مفهومی با برخی رویکردهای جدید انجام می‌شود. در کار [۱]، یک نسخه بهبودیافته از الگوریتم ARO برای تولید آرایه‌های پوششی ارائه شده است که تمرکز آن عمدتاً بر تولید آرایه‌های یکنواخت بدون مدل‌سازی صریح قیود است، در حالی که روش حاضر با بهره‌گیری از ROBDD، امکان اعمال قیود ساختاری را پیش از فرایند بهینه‌سازی فراهم می‌کند. در [۵] و [۸] نیز رویکردهای سه مرحله‌ای و ممتیک برای تولید آرایه‌های با قدرت بالا پیشنهاد شده‌اند که بیشتر بر بهبود کیفیت جستجو در فضای ترکیبیات تمرکز دارند و کاهش تعداد متغیرها در سطح مدل را هدف قرار نمی‌دهند. مقاله [۱۹] با استفاده از الگوریتم‌های فرابتکاری برای ساختاردهی بهینه آرایه‌های پوششی در حوزه دفاعی، نشان می‌دهد که ترکیب چند الگوریتم می‌تواند نتایج بهتری ایجاد کند. همچنین در [۲۹] رویکردی مدل محور برای خودکارسازی

UNCORRECTED PROOF

تولید آرایه‌های پوششی ارائه شده است که در آن پارامترها و مقادیر به کمک مدل چکینگ GROOVE استخراج می‌شوند و سپس ترکیب الگوریتم‌های BBO و GA برای تولید مجموعه آزمون کمینه و کارا به کار می‌رود؛ با این حال تمرکز اصلی آن کار بر خودکارسازی استخراج پارامترها و بهبود کارایی جستجو است، در حالی که روش پیشنهادی این مقاله با ادغام ROBDD و BBO، هم زمان دو هدف کاهش تعداد متغیرهای مدل و کاهش اندازه مجموعه آزمون را محقق کرده و در مقایسه با رویکردهای مذکور، ساختاری منسجم برای پشتیبانی از آرایه‌های پوششی با محدودیت ارائه می‌دهد.

۸-۲- محدودیت‌های روش پیشنهادی

با وجود برتری‌های عددی و کاهش محسوس تعداد نمونه‌های آزمون نسبت به روش‌های موجود، روش پیشنهادی با چند محدودیت همراه است:

- اندازه گراف ROBDD به‌شدت به نحوه مدل‌سازی و ترتیب متغیرها وابسته است؛ در برخی سیستم‌های بزرگ‌تر، امکان انفجار اندازه گراف وجود دارد که می‌تواند هزینه محاسباتی روش را افزایش دهد.
- پیاده‌سازی فعلی عمدتاً برای ساختارهای دودویی و پیکربندی‌های استاندارد با تعداد پارامتر و مقادیر محدود ارزیابی شده است و کارایی آن روی آرایه‌های مختلط با تعداد سطوح بزرگ‌تر هنوز به طور سیستماتیک بررسی نشده است.
- الگوریتم BBO ذاتاً تصادفی است و به تنظیم مناسب پارامترها (مانند نرخ مهاجرت، تعداد تکرارها و اندازه جمعیت) حساس است؛ در صورت تنظیم نامناسب، همگرایی ممکن است کند شود یا در بهینه‌های محلی متوقف گردد.
- ارزیابی تجربی روی یک مطالعه موردی اصلی (سیستم خاموش‌سازی بویلر) و چند پیکربندی استاندارد انجام شده است؛ برای تعمیم نتایج، آزمایش روی دامنه‌های صنعتی متنوع‌تر ضروری است.
- در نسخه فعلی، محدودیت‌ها در سطح مدل علت و معلول و ROBDD اعمال می‌شوند و پشتیبانی از انواع پیچیده‌تر قیود (مثلاً قیود زمانی یا وابستگی‌های پویا) نیازمند توسعه بیشتر چارچوب است.

۹- نتیجه‌گیری و کارهای آتی

در این پژوهش، یک روش جدید برای تولید مجموعه آزمون بهینه با پوشش ۱۰۰ درصد ارائه شد که ترکیب گراف‌های ROBDD با الگوریتم هوش مصنوعی BBO را به کار گرفت. این روش توانست در مقایسه با سایر روش‌های موجود، از جمله روش‌های مبتنی بر هوش مصنوعی مانند PSO-ROBDD و روش‌های غیر هوش مصنوعی همچون Jenny و TConfig، عملکرد بهتری از خود نشان دهد. در تمامی مقاطع ارزیابی، روش پیشنهادی قادر بود تعداد نمونه‌های آزمون را به طور قابل توجهی کاهش دهد، در حالی که پوشش ۱۰۰ درصدی را حفظ کرده و از لحاظ تعداد متغیرها نیز بهینه‌تر عمل کرد. این ویژگی‌ها نشان‌دهنده بهره‌وری بالا، کاهش هزینه‌ها و زمان اجرای فرآیند آزمون‌ها است.

روش پیشنهادی همچنین با استفاده از گراف‌های علت و معلولی و تبدیل آنها به گراف‌های ROBDD، به شناسایی علل غیرضروری پرداخته و موجب کاهش چشمگیر تعداد متغیرها شد. این بهینه‌سازی نه تنها در کاهش هزینه‌ها و زمان اجرا مؤثر است، بلکه به شفافیت و کارایی سیستم‌های پیچیده در تست‌های نرم‌افزاری کمک می‌کند.

برای بهبود بیشتر این پژوهش، می‌توان به بررسی و پیاده‌سازی روش‌های جدید در ترکیب با الگوریتم‌های هوش مصنوعی پیشرفته‌تر، مانند الگوریتم‌های یادگیری عمیق، برای بهینه‌سازی تولید مجموعه‌های آزمون با پوشش بیشتر و دقت بالاتر پرداخت. همچنین، استفاده از الگوریتم‌های تکاملی جدید [۳۰]، برای بررسی مقایسه‌های بیشتر و بهینه‌سازی فرآیند می‌تواند مفید باشد. علاوه بر این، گسترش این روش‌ها به سایر انواع سیستم‌ها و دامنه‌های پیچیده‌تر می‌تواند منجر به کاربردهای جدید و مفیدی در صنعت نرم‌افزار و سیستم‌های تعبیه‌شده شود.

یکی دیگر از راه‌های تحقیقاتی، بررسی تأثیر زیرساخت‌های مختلف سخت‌افزاری و نرم‌افزاری بر کارایی این الگوریتم‌ها و بهینه‌سازی‌های بیشتر در شرایط واقعی است. همچنین، می‌توان به‌کارگیری این الگوریتم‌ها در پروژه‌های صنعتی و مقیاس‌های بزرگتر، به ویژه در سیستم‌های حساس و پیچیده، را مد نظر قرار داد تا ارزیابی دقیق‌تری از کارایی و قابلیت‌های عملی آن‌ها به دست آید.

به طور کلی با توجه به نتایج به‌دست‌آمده، مسیرهای زیر برای تحقیقات آتی پیشنهاد می‌شود:

UNCORRECTED PROOF

- توسعه روش پیشنهادی برای آرایه‌های پوششی مختلط و با قدرت‌های متغیر، به‌طوری که بتوان مقادیر مختلف برای هر پارامتر و سطوح تعامل متفاوت را به طور هم‌زمان پشتیبانی کرد.
- ترکیب چارچوب ROBDD–BBO با سایر الگوریتم‌های تکاملی و فراابتکاری جدید (مانند نسخه‌های پیشرفته الگوریتم تفاضلی، الگوریتم‌های مبتنی بر یادگیری و بهینه‌سازی‌های ریاضی جدید) به منظور بهبود سرعت همگرایی و کیفیت راه‌حل‌ها.
- پیاده‌سازی نسخه‌های موازی و توزیع‌شده از الگوریتم پیشنهادی بر بسترهای چندهسته‌ای و خوشه‌های محاسباتی، برای به‌کارگیری در پیکربندی‌های بسیار بزرگ و پیچیده.
- ارزیابی روش بر روی مجموعه‌ای متنوع از سیستم‌های واقعی در حوزه‌های مختلف (نظیر سیستم‌های نهفته، سامانه‌های هوافضا و سامانه‌های با قابلیت اطمینان بالا) به منظور بررسی عملی بودن و تعمیم‌پذیری نتایج.
- بررسی دقیق‌تر تأثیر تنظیمات پارامترهای BBO و راهکارهای سازگارسازی پویا (adaptive) برای کاهش حساسیت روش به پارامترها و افزایش پایداری عملکرد در اجراهای متعدد.

تعارض منافع

نویسندگان این مقاله تأیید می‌کنند که هیچ‌گونه تعارض منافع مالی، مشاوره‌ای، نهادی یا دیگر روابطی که ممکن است منجر به سوگیری یا تضاد منافع در این پژوهش گردد، وجود ندارد.

تاییدیه اخلاقی

نویسندگان متعهد می‌شوند که مطالب این مقاله را در هیچ مجله دیگری به چاپ نرسانده‌اند.

مشارکت های نویسندگان

خانم اعظم امرایی مسئول پیاده‌سازی و نوشتن نسخه اولیه مقاله بوده‌اند، در حالی که سجاد اسفندیاری نقش ویرایش، هدایت علمی مقاله و ارائه ایده اولیه را بر عهده داشته‌اند.

منابع مالی

این پژوهش هیچ‌گونه منابع مالی از نهادها یا سازمان‌های خارجی دریافت نکرده است.

مراجع

[1] Pira, E., and M. Khodizadeh-Nahari. "Combinatorial t-way test suite generation using an improved asexual reproduction optimization algorithm." Applied Soft Computing 150 (2024).

[2] Wu, H., C. Nie, F. C. Kuo, H. Leung, and C. J. Colbourn. "A discrete particle swarm optimization for covering array generation." IEEE Transactions on Evolutionary Computation 19, no. 4 (2015): 575-591.

[3] Esfandyari, S., L. Yousofvand, E. Pira, and V. Rafe. "Optimal production of the test suite by the combinatorial testing method by applying changes in the gravitational search algorithm for the uniform strength cover array." Tabriz Journal of Electrical Engineering 54, no. 3 (2024): 269-279 (in Persian).

[4] Yousofvand, L., S. Soleimani, V. Rafe, and A. Nikanjam. "Graph neural networks for precise bug localization through structural program analysis." Automated Software Engineering 33, no. 1 (2026).

[5] Pira, E., V. Rafe, and S. Esfandyari. "A three-phase approach to improve the functionality of t-way strategy." Soft Computing (2023): 1-21.

[6] Guo, X., X. Song, J. Zhou, F. Wang, K. Tang, and Z. Wang. "A memetic algorithm for high-strength covering array generation." IET Software 17, no. 4 (2023): 538-553.

[7] Yousofvand, L., M. B. Dowlatshahi, and M. P. Beiranvand. "Pneumonia detection in chest X-ray images using convolutional neural network and fuzzy VIKOR." Iranian Journal of Fuzzy Systems 22, no. 5 (2025): 159-179.

[8] Esfandyari, S., and V. Rafe. "GALP: A hybrid artificial intelligence algorithm for generating covering array." Soft Computing 25 (2021): 7673-7689.

[9] Esfandyari, S., and V. Rafe. "A tuned version of genetic algorithm for efficient test suite generation in interactive t-way testing strategy." Information and Software Technology 94 (2018): 165-185.

[10] Esfandyari, S., and V. Rafe. "Extracting combinatorial test parameters and their values using model checking and evolutionary algorithms." Applied Soft Computing 91 (2020): 1-19.

[11] Abbasi, Z., S. Esfandyari, and V. Rafe. "Covering array generation using teaching-learning-based optimization algorithm." Tabriz Journal of Electrical Engineering 48, no. 1 (2018): 161-171 (in Persian).

UNCORRECTED PROOF

[12] Esfandyari, S., and V. Rafe. "Using the particle swarm optimization algorithm to generate the minimum test suite in covering array with uniform strength." Soft Computing Journal 8, no. 2 (2021): 66-79 (in Persian).

[13] Ahmed, B. S., T. Sh. Abdulsamad, and M. Y. Potrus. "Achievement of minimized combinatorial test suite for configuration-aware software functional testing using the cuckoo search algorithm." Information and Software Technology 66 (2015): 13-29.

[14] Esfandyari, S., and V. Rafe. "A hybrid solution for software testing to minimum test suite generation using hill climbing and bat search algorithms." Tabriz Journal of Electrical Engineering 46, no. 3 (2016): 25-35 (in Persian).

[15] Torres-Jimenez, J., and E. Rodriguez-Tello. "Simulated annealing for constructing binary covering arrays of variable strength." In Proceedings of the Congress on Evolutionary Computation, Barcelona, Spain, July 2010.

[16] Torres-Jimenez, J., and C. Perez-Torres. "A greedy algorithm to construct covering arrays using a graph representation." Information Sciences 477 (2019): 234-245.

[17] Kalaee, A., and V. Rafe. "An optimal solution for test case generation using ROBDD graph and PSO algorithm." Quality and Reliability Engineering International 32, no. 7 (2016): 2263-2279.

[18] Simon, D. "Biogeography-based optimization." IEEE Transactions on Evolutionary Computation 12, no. 6 (2008): 702-713.

[19] Esfandyari, S., D. Giveki, and M. Farokhzadi. "A novel approach to enhancing defense: metaheuristic algorithms for optimal structuring of covering arrays and efficient test suite generation." Aerospace Defense 2, no. 4 (2024): 58-77 (in Persian).

[20] Esfandyari, S., and V. Rafe. "Correction to: GALP: a hybrid artificial intelligence algorithm for generating covering array." Soft Computing (2021).

[21] Avila-George, H., J. Torres-Jimenez, L. Gonzalez-Hernandez, and V. Hernández. "Metaheuristic approach for constructing functional test-suites." IET Software 7, no. 2 (2013): 104-117.

[22] Pira, E., V. Rafe, and S. Esfandyari. "Minimum covering array generation using success-history and linear population size reduction based adaptive differential evolution algorithm." Tabriz Journal of Electrical Engineering 52, no. 2 (2022): 77-89 (in Persian).

[23] Torres-Jimenez, J., and E. Rodriguez-Tello. "New bounds for binary covering arrays using simulated annealing." Information Sciences 185, no. 1 (2012): 137-152.

[24] Gonzalez-Hernandez, L., N. Rangel-Valdez, and J. Torres-Jimenez. "Construction of mixed covering arrays of variable strength using a tabu search approach." In International Conference on Combinatorial Optimization and Applications, Berlin, Heidelberg, 2010.

[25] Zamli, K. Z., B. Y. Alkazemi, and G. Kendall. "A tabu search hyper-heuristic strategy for t-way test suite generation." Applied Soft Computing 44 (2016): 57-74.

[26] Ahmed, B. S., K. Z. Zamli, and C. P. Lim. "Application of particle swarm optimization to uniform and variable strength covering array construction." Applied Soft Computing 12, no. 4 (2012): 1330-1347.

[27] Alsewari, A., A. Rahman, and K. Z. Zamli. "Design and implementation of a harmony-search-based variable-strength t-way testing strategy with constraints support." Information and Software Technology 54, no. 6 (2012): 553-568.

[28] Rodriguez-Tello, E., and J. Torres-Jimenez. "Memetic algorithms for constructing binary covering arrays of strength three." In International Conference on Artificial Evolution (Evolution Artificielle), Berlin, Heidelberg, 2009.

[29] Esfandyari, S., D. Giveki, A. Rastegarnia, A. Farzamnia, and H. Zheng. "Efficiently automating covering array generation: a model checking and metaheuristic algorithm approach." IEEE Access 13 (2025): 122804-122821.

[30] Giveki, D., J. Ebrahimi, and M. Sarshar. "Proposing an improved version of the bat algorithm." Journal of Modeling in Engineering 22, no. 79 (2024): 267-279 (in Persian).

UNCORRECTED PROOF