A Local Search Approach for Batch Repair With Configuration Neighbor and Sub-Degree in Circuit System

MENG LIU1,2, DANTONG OUYANG1,2, NAIYU TIAN1,2, AND HUA LI3
1College of Computer Science and Technology, Jilin University, Changchun 130012, China
2Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun 130012, China
3Department of Abdominal Ultrasound, The First Hospital of Jilin University, Changchun 130021, China

Corresponding author: Hua Li (jlulihua@163.com)

This work was supported by the National Natural Science Foundation of China under Grant 61672261 and Grant 61872159.

Abstract The goal of automated troubleshooting is to diagnose and fix a system by repairing its abnormal behaviors, and this process has wide application. Heuristic search is an efficient way to solve troubleshooting problems, such as the batch repair problem (BRP), and heuristic search aims to minimize the costs of repairing a circuit system. The local search method is an efficient search approach and has very good performance in many areas. The key to this method is the modeling of “neighbor” and the search strategy. This paper proposes a local search approach for batch repair with configuration neighbors and sub-degree (CNASD). First, we define configuration neighbors in the BRP. Second, we propose three strategies to guide the search process, namely, best diagnosis with the highest probability component (BDHC), configuration neighbor with aspiration (CNA) and minimal sub-degree. Experimentally, compared with state-of-the-art batch repair algorithms that use heuristic search, our algorithm performs better in terms of the average / maximum repair cost and average / maximum runtime (decreased by up to one order of magnitude) on ISCAS’85 and ITC’99 systems, which are the standard combinational circuit systems.

Index Terms Local search, model-based diagnosis, batch repair, configuration neighbor, sub-degree.

I. INTRODUCTION

In general, diagnosing a system refers to identifying the root causes of encountered errors. A diagnostic problem arises when an inconsistency appears between the predicted behavior and observed behavior. Diagnosis algorithms return a set of possible diagnoses. The components with abnormal behavior in a diagnosis can explain the inconsistency. As the system’s scale increases, diagnosis problems often become more common, and the maintenance costs potentially increase. Hence, automated diagnosis has been studied for several decades, with a great deal of development and successful applications in circuit diagnosis [1]–[9], spacecraft [10], satellite decision support systems [11], the automotive industry [12], spreadsheets [13] and nonlinear systems [14], [15].

The diagnosis problem also accompanies the emergence of troubleshooting problems. The goal of solving a troubleshooting problem is to incur the minimal cost until the system is fixed. Note that, in this paper, “repair” is used for one or more components, and “fix” is used for the entire system. Most previous work on troubleshooting assumed that components are repaired one at a time [16]–[19]. When the repair overhead is more expensive than the component repair cost, this approach can be wasteful. It may be more effective to repair more than one component in a single repair action. We call this approach the batch repair problem (BRP). Unlike traditional troubleshooting problems, the costs incurred in the BRP include repair overhead costs and component repair costs. The component repair cost is the cost of the repair of a single component. The repair overhead cost denotes additional repair costs, for example, due to repairing one component at a time in a system with multiple faults. This approach is obviously wasteful in the BRP. Each repair action requires additional costs, i.e., repair overhead costs. However, it is more efficient to repair all faults in a single repair action. Hence, the main problem of the BRP is how to select multiple components for a single repair action.
Stern et al. propose wasted cost utility functions [20] to guide this process in circuit systems. The wasted cost includes false positive costs and false negative costs. The false positive costs are the costs that appear when some repaired components are not truly faulty. False negative costs are the overhead costs incurred by the further repair actions. Batch repair algorithms use different utility functions according to the optimistic or pessimistic expected cost of false negatives. Moreover, these algorithms compare the costs with two baseline repair algorithms inspired by previous work on test planning [21], namely, “Best Diagnosis” and “Highest Probability”. These baseline repair algorithms do not consider batch repair. Hence, these algorithms are used to evaluate the feasibility of batch repair algorithms. The “Best Diagnosis” repair algorithm chooses to repair a single component according to the most likely diagnosis. The “Highest Probability” selects the component to repair that is most likely to be faulty, as computed by the system’s health state [22]. The experiments show that batch repair has excellent performance.

Subsequently, Shinitzky et al. [1] propose the enhanced wasted cost utility function and use heuristic search algorithms in a circuit system. The experimental results show that the enhanced wasted cost utility function outperforms previous utility functions from [20]. However, the universal winner does not appear in those heuristic search algorithms.

More importantly, for exploring the search space, Shinitzky et al. [1] and Stern et al. [20] propose a union-based search and a powerset-based search. Union-based search considers the union set of multiple diagnoses, and a powerset-based search considers the union set of multiple components in each search iteration process. In fact, a powerset-based search ignores the reasonability of a diagnosis, i.e., whether the diagnosis can explain the abnormal behavior of the current system. More importantly, the BRP’s goal of incurring minimal costs has a critical precondition. That is, the system must have been fixed after the batch repair sequence. Hence, it is not reasonable to only consider the union of the components when minimizing cost. For a union-based search, a redundant component (that is not truly faulty) may exist without being detected due to considering one comprehensive diagnosis at a time. Hence, there is a key problem of how to fix a system sooner with batch repair and subsequently obtain minimal costs.

Previous work indicates that it is difficult to model the BRP as a Markov decision process [1]. As an alternative, the BRP can be modeled as a combinatorial optimization problem, in which the goal is to search for the optimal batch repair action.

Local search is a heuristic algorithm to solve optimization problems. Local search has very good performance in many areas [23]–[31]. As a popular approach to solving NP-hard combinatorial problems, local search is well known for its ability to quickly find a good-quality solution. The main idea of local search is to start with an initial solution and iteratively modify it according to certain heuristics. The key to this is the modeling of neighbors and the heuristics.

In this paper, our first contribution is a local search method for batch repair with configuration neighbors and sub-degree (CNASD). According to the relationship between the current batch repair sequence and the set of diagnoses, we define the configuration neighbor of the current batch repair sequence in the BRP. This approach is helpful for modifying the solution in local search. Our second contribution is a set of three strategies. We propose the strategies of: best diagnosis with the highest probability component (BDHC), configuration neighbor with aspiration (CNA) and minimal sub-degree. In the local search approach, the BDHC strategy is used to obtain the initial batch repair sequence. Then, the CNA and minimal sub-degree strategies are used to modify the sequence (sometimes the BDHC is also used). Our third contribution is a hybrid search space. As we described before, for the search space, the previous approach to batch repair considers either the union of diagnoses or the union of components. Because the precondition of the BRP is that the system must have been fixed, only considering diagnoses or components is not reasonable for obtaining minimal costs. Only considering diagnoses may lead to redundant components in the batch repair sequence. Only considering components ignores the reasonable premise that diagnosis can explain the current abnormal behavior of the system. Specifically, this approach blindly pursues low costs but ignores the precondition of the BRP (the system is fixed). Hence, the three strategies we propose aim to fix the system earlier and then obtain minimal costs using a hybrid search space. We search an entire diagnosis for a batch repair sequence with the BDHC strategy, and we search components with the CNA and minimal sub-degree strategies. We consider the reasonability of diagnosis and the minimization of redundant components.

We demonstrate our local search approach on a combinational Boolean circuit from the standard ISCAS‘85 and ITC’99 benchmarks. The results show that with 1-5 faults, we are able to find a lower average / maximum cost incurred before the system is fixed, compared with state-of-the-art batch repair algorithms with heuristic search [1], and we reduce the average / maximum runtime by up to one order of magnitude.

The paper is organized as follows. Section 2 introduces some preliminaries. Section 3 presents three strategies and some notions. Then, we describe the CNASD approach and the details of the main process. Section 4 compares the CNASD approach with state-of-the-art batch repair algorithms with heuristic search [1]. Finally, we conclude the paper in Section 5.

II. PRELIMINARIES

First, we show the whole workflow of the model-diagnosis-troubleshooting process in Figure 1.

In model-based diagnosis (MBD), a model of the system is first built, namely, a triple \((SD, Comps, Obs)\) [32], [33]. The system description is denoted as \(SD\), the set of components are denoted as \(Comps\), and a given observation is denoted...
as Obs. Then, the predicted behavior of the system is obtained by following the model. Recently, MBD algorithms have focused on a satisfiability- or maximum satisfiability- (SAT or MaxSAT, respectively) based approach [2], [4], [34]–[37] because of the rapid development of SAT and MaxSAT technology [23], [25], [38]. As shown in Figure 1, the model process obtains a (weighted) conjunctive normal form ((W)CNF) file based on gate logic and knowledge of discrete mathematics [39], [40]. The numeric variables are used to denote components, inputs and outputs. Obs and Comps are modeled as unit clauses, and SD is modeled based on gate logic and knowledge of discrete mathematics. The details of the circuit model process are shown in [39], [40].

An MBD problem arises when an inconsistency appears between the observed behavior of the system and the predicted behavior. That is, the system has abnormal behavior. A troubleshooting problem, in general, plans a sequence of actions to fix an abnormally behaving system.

Recently, batch repair has been an efficient way to solve troubleshooting problems. The BRP arises when an MBD problem arises, i.e., the system’s behavior is abnormal.

A description of the diagnosis problem and the related concepts can be found in [32], [33]. A unary predicate Ab indicates an unhealthy component \( c \in \text{Comps} \), i.e., \( \text{Ab}(c) \) is true when component \( c \) has abnormal behavior.

Definition 1 (Diagnosis): Given an MBD problem \( \langle \text{SD}, \text{Comps}, \text{Obs} \rangle \), \( \Delta \subseteq \text{Comps} \) is a diagnosis when \( \text{SD} \cup \text{Obs} \cup \{ \text{Ab}(c) | c \in \Delta \} \cup \{ \neg \text{Ab}(c) | c \in \text{Comps} \setminus \Delta \} \) is consistent.

In addition, \( \Delta \) is a subset-minimal diagnosis when any \( \Delta' \subset \Delta \) is not a diagnosis. \( \Delta \) is a cardinality-minimal diagnosis if no diagnosis \( \Delta' \) exists such that \( |\Delta'| < |\Delta| \).

In BRP, the model-based diagnosis engine (MBDE) returns a set of diagnoses \( \mathcal{D} \) and their likelihoods \( p(d) \), where a diagnosis \( d \) is a set of components that may have been faulty, and the likelihood \( p(d) \) is the likelihood of \( d \).

Definition 2 (Health State): A health state \( H[i] \) is an estimate of the likelihood that component \( c_i \in \text{Comps} \) is faulty. A health state can be obtained via a sequence of diagnoses and their likelihoods. The likelihood for each component can be obtained as \( H[i] = \sum_{\Delta \in \Omega} p(\Delta) \cdot \beta_{c_i \in \Delta} \), where \( \beta_{c_i \in \Delta} \) is assigned 1 when \( c_i \in \Delta \) and 0 otherwise.

We show an example of health states in Table 1. Given a system with 6 components, there are 5 diagnoses \( d_1 = \{c_1, c_2, c_4\}, d_2 = \{c_1, c_3\}, d_3 = \{c_2, c_5\}, d_4 = \{c_3, c_5\}, d_5 = \{c_1, c_5, c_6\} \) with corresponding likelihoods 0.1, 0.2, 0.3, 0.2, and 0.2. The likelihood for \( c_1 \) is \( H[1] = 0.1 \cdot 1 + 0.2 \cdot 1 + 0.3 \cdot 0 + 0.2 \cdot 0 + 0.2 \cdot 1 = 0.5 \), and we obtain \( H[5] = 0.7 \) similarly.

Through a SAT- or MaxSAT-based diagnosis algorithm, we can obtain diagnosis results via the CNF or WCNF file,
as shown in Figure 1, including the health state, diagnosis and score. The subsequent troubleshooting process requires these diagnosis results to obtain repair sequences.

Specifically, when an MBD problem arises, at least one component must be repaired. Thus, repair actions are needed to fix the system. A description of the repair action and related concepts can be found in [20].

Definition 3 (Repair Action): A repair action aims to repair a set of components \( \gamma \subseteq \text{Comps} \), denoted by \( \text{Repair}(\gamma) \).

Note that the components in \( \gamma \) have normal behavior after the repair action. Then, it is verified whether the system has been fixed via the comparing observed system outputs and the expected outputs of a healthy system. In this paper, the components \( \gamma \) are called the batch repair sequence.

More importantly, costs will be incurred after repairing some components, and they are composed of the repair overhead and component repair costs. The repair overhead is denoted as \( \text{cost}_{\text{repair}} \), and the repair cost of a component \( c \in \text{Comps} \) is denoted by \( \text{cost}_{c} \).

Definition 4 (Repair Costs): After a repair action, the cost incurred is \( \text{cost}(\text{Repair}(\gamma)) = \text{cost}_{\text{repair}} + \sum_{c \in \gamma} \text{cost}_{c} \), where \( \text{cost}_{\text{repair}} > 0 \) and \( \text{cost}_{c} > 0 \).

The BRP algorithm is a way to solve troubleshooting problems, i.e., to fix the system. To minimize the total repair cost is the task of the BRP.

Definition 5 (System Repair Likelihood): The system repair likelihood \( \text{SystemRepair}(\gamma) \) (\( \gamma \subseteq \text{Comps} \)) is the probability that \( \text{Repair}(\gamma) \) would fix the system.

The system repair likelihood can be obtained via a sequence of repaired components \( \gamma \) and a set of diagnoses \( \Omega \). The likelihood for each component can be obtained as \( \text{SystemRepair}(\gamma) = \sum_{\Delta \in \Omega} p(\Delta) \).

Previous works modeled the BRP as a combinatorial optimization problem and used the heuristic search method to solve it [1], [20]. They use heuristic search methods, i.e., A* variant, hill climbing (HC) and depth-limited breadth-first search. To find the minimum total repair cost, utility functions are proposed as heuristic strategies. These functions consist of false positive costs \( \text{cost}_{FP} \), false negative costs \( \text{cost}_{FN} \) and future false positive costs \( \text{cost}_{FFP} \). \( \text{cost}_{FP} \) denotes the costs incurred by repairing components that are not truly faulty, i.e., it indicates whether the current repair actions involve some non-faulty components. \( \text{cost}_{FN} \) gives the overhead costs incurred by succeeding repair actions. \( \text{cost}_{FFP} \) denotes the costs that will be wasted in the future when repairing components that should not have been repaired. Then, \( \text{cost}_{FFP} \) is estimated according to equation 4. These cost functions are as follows.

\[
\text{cost}_{FP}(\gamma) = \sum_{c \in \gamma} (1 - \text{cost}(c)) \quad (1)
\]

\[
\text{cost}_{FP}^o(\gamma) = \text{cost}_{\text{repair}} \quad (2)
\]

\[
\text{cost}_{FP}^o(\gamma) = \text{cost}_{\text{repair}} + \sum_{i \in \gamma} \text{H}(i) \quad (3)
\]

\[
\text{cost}_{FFP}(\gamma) = \sum_{c \in \gamma} (1 - \text{cost}(c)) \quad (4)
\]

Note that \( \text{cost}_{FN} \) considers two cases. The optimistic case \( \text{cost}_{FN}^o(\gamma) \) only needs one additional repair action. In the pessimistic case, \( \text{cost}_{FN}^p(\gamma) \), one faulty component is repaired per repair action in the future. These are two kinds of extreme scenarios.

Stern et al. [20] proposed two utility functions, namely, \( C_{W_{\text{Opt}}} \) and \( C_{W_{\text{Pess}}} \). \( C_{W_{\text{Pess}}} \) has better results. Subsequently, the enhanced wasted cost utility function \( C_{W_{\text{CFP}}} \) is proposed for use in search algorithms that use A* variant, hill climbing (HC) and depth-limited breadth-first search [1]. Moreover, compared with \( C_{W_{\text{Opt}}} \) and \( C_{W_{\text{Pess}}} \), \( C_{W_{\text{CFP}}} \) performs better. These utility functions are as follows.

\[
C_{W_{\text{Opt}}} = \text{cost}_{\text{FP}}(\gamma)
+ (1 - \text{SystemRepair}(\gamma)) \cdot \text{cost}_{\text{FN}}^p(\gamma) 
(5)
\]

\[
C_{W_{\text{Pess}}} = \text{cost}_{\text{FP}}(\gamma)
+ (1 - \text{SystemRepair}(\gamma)) \cdot \text{cost}_{\text{FN}}^p(\gamma) 
(6)
\]

\[
C_{W_{\text{CFP}}} = \text{cost}_{\text{FP}}(\gamma)
+ (1 - \text{SystemRepair}(\gamma)) \cdot \text{cost}_{\text{FN}}^p(\gamma) 
+ \text{cost}_{\text{FFP}}(\gamma) 
(7)
\]

More importantly, Shinitzky et al. [1] propose that the BRP can be modeled as a combinatorial optimization problem of searching for the optimal batch repair action. However, these authors do not find the universal winner with A*, HC or depth-limited breadth-first search. Therefore, there is much room for improvement here.

### III. A LOCAL SEARCH APPROACH FOR BATCH REPAIR

In this section, we propose a novel local search approach for BRP, namely, CNASD. The key idea of local search is defining “neighbor” and choosing a heuristic strategy. We first introduce formal notions. Next, we present three strategies to be used in our approach: BDHC, CNA and minimal sub-degree. In addition, we describe how we obtain an optimal batch repair sequence via CNASD.

### A. NOTIONS

To introduce our CNASD method, we first define some concepts.

Definition 6 (Configuration Neighbor): Given a set of components \( \gamma \) in a batch repair sequence and a set of diagnoses \( \Omega = \{\Delta_1, \Delta_2, \ldots, \Delta_m\} \), the configuration neighbors of \( \gamma \), denoted by \( CN(\gamma) \), consist of the components \( v \) that appear

| TABLE 1. An example for diagnoses and health state. |
| --- |
| \( d_1 \) | \( d_2 \) | \( d_3 \) | \( d_4 \) | \( d_5 \) | \( H[\gamma] \) |
| \( c_1 \) | 1 | 1 | 0 | 0 | 1 | 0.5 |
| \( c_2 \) | 0 | 1 | 0 | 0 | 0 | 0.4 |
| \( c_3 \) | 0 | 1 | 0 | 1 | 0 | 0.4 |
| \( c_4 \) | 1 | 0 | 0 | 0 | 1 | 0.1 |
| \( c_5 \) | 0 | 0 | 0 | 1 | 1 | 0.7 |
| \( c_6 \) | 0 | 0 | 0 | 0 | 0 | 0.2 |
| \( p(\cdot) \) | 0.1 | 0.2 | 0.3 | 0.2 | 0.2 | 0.2 |
in the same diagnosis $\Delta \in \Omega$ as each component $u \in \gamma$, i.e., $CN(\gamma) = \bigcup_{j=1}^{m} \Delta_j \setminus \gamma (\Delta_j \cap \gamma \neq \emptyset)$. That is, for a component $u \in \gamma$, if components $v \notin \gamma$ and $u$ appear in the same diagnosis, we say that component $v$ is a configuration neighbor of $\gamma$, i.e., $v \in CN(\gamma)$.

We take Table 1 as an example. If the current batch repair sequence $\gamma = \{c_1, c_3\}$, then $CN(\gamma)$ consists of $c_1, c_2$ and $c_6$, i.e., $CN(\gamma) = \{c_1, c_2, c_6\}$. Obviously, $c_1 \notin \gamma$ and $c_3 \notin \gamma$ are in the same diagnosis $d_2$. The components $c_5 \in \gamma, c_1 \notin \gamma$ and $c_6 \notin \gamma$ are in the same diagnosis $d_3$.

Definition 7 (Sub-Degree): Given a set of components $\gamma$ and a set of diagnoses $\Omega$, the sub-degree of a diagnosis $\Delta \in \Omega$ is the cardinality of the difference of $\gamma$ and $\Delta$, denoted by $subdegree(\Delta) = |\Delta \setminus \gamma|$, i.e., $|\Delta| - |\gamma \cap \Delta|$.

Take Table 1, for example. If $\gamma = \{c_5\}$, the sub-degree of $d_3$ is 1, and $subdegree(d_3) = 2$. The sub-degree of a diagnosis shows the “distance” from the set of components $\gamma$ to this diagnosis, to some extent.

B. THREE STRATEGIES FOR BATCH REPAIR

As we described before, union-based search is better than powerset-based search in terms of solution quality. However, union-based search also has defects. Union-based search means we need to repair the union of multiple comprehensive diagnoses. In this case, the batch repair sequence may have some unnecessary components that are not true faults in the system. Powerset-based search ignores the reasonability of a diagnosis, i.e., whether the diagnosis can explain abnormal system behavior. Moreover, a unitary search space is not reasonable. Hence, we propose three strategies to solve the BRP in our heuristic search algorithm. The details are as follows.

1. Strategy 1 (Best Diagnosis With the Highest Probability Component, BDHC): Given a set of diagnoses $\Omega = \{\Delta_1, \Delta_2, \ldots, \Delta_m\}$ with corresponding likelihoods $p(\Delta_j)(j \in [1, m])$, components $Comps = \{c_1, c_2, \ldots, c_n\}$ and health state of each component $H_i(i \in [1, n])$, the BDHC strategy chooses a diagnosis $\Delta \in \Omega$ with maximal $p(\Delta)$, which contains the $c_i$ with highest $H[i]$, i.e., $\Delta = \max_{\Omega}^{\Delta} (p(\Delta_j))$ contains $c_i \in \Delta(\max_{Comps}^{\Delta} H[i])$.

The MBDE returns a set of diagnoses and their likelihoods. The probability for a component, i.e., the health state, is obtained via aggregating information on diagnoses. The BDHC strategy aims to find the best diagnosis (i.e., that with the maximal likelihood) that includes the highest probability component.

Considering Table 1 as an example, $c_5$ has the highest probability, and $d_3, d_5$ both contain $c_5$. We obtain $d_3$ as the best diagnosis with the highest probability component because $d_3$ has a higher likelihood than $d_5$ ($p(d_3) = 0.3 > p(d_5) = 0.2$).

2. Strategy 2 (Configuration Neighbor With Aspiration, CNA): Given a set of components $\gamma(\gamma(\gamma) > 1)$, the CNA strategy selects a component $c_i \in CN(\gamma)$ with highest $H[i]$, i.e., $c_i = \max_{i}^{\gamma} H[i]$

For a set of components $\gamma$, its configuration neighbors may be not be unique if they exist, i.e., $|CN(\gamma)| > 1$. Hence, we combine an aspiration mechanism with the configuration neighbor to make it more flexible. The CNA strategy selects a component $c_i$ with the highest $H[i]$ from $CN(\gamma)$. Typically, the CNA strategy is used in the greedy mode to decrease blind unreasonable greedy search.

3. Strategy 3 (Minimal Sub-Degree): Given a set of components $Comps_{CNA} = \{\max_{i}^{\gamma} H[i]\}$ in which each component $c_i \in Comps_{CNA}$ satisfies strategy 2 and a set of diagnoses $\Omega_k = \{\Delta \cap Comps_{CNA} \neq \emptyset\}$, the minimal sub-degree strategy selects a component $c_i \in Comps_{CNA}$ with minimal sub-degree, i.e., $c_i = \max_{i}^{\min_{\Omega_k} subdegree(\Delta) \cap \gamma} H[i]$

Strategy 3 is based on strategy 2. As in strategy 2, the component with highest likelihood from $CN(\gamma)$ is chosen. However, there may be more than one component that satisfies this requirement. The diagnoses including these components are taken into account. Hence, we utilize the sub-degree to supplement our search algorithm according to strategy 3 in the two cases that follow.

(Case a). The diagnosis $\Delta (\exists c_i \in \Delta | c_i \in Comps_{CNA})$ with minimal sub-degree is taken into account. Then, in $\Delta \setminus \gamma$, the component $c_i$ with highest $H[i]$ is chosen according to strategy 3, i.e., $c_i = \max_{\gamma}^{\Delta \setminus \gamma} H[i]$. This choice enables our approach to select as much of the whole diagnosis as possible.

(Case b). There is sometimes more than one diagnosis with minimal sub-degree. The latest-visited diagnosis $\Delta_{Latest \text{visited}}$ (which contains the latest selected component) is taken into account. Then, the component with the highest likelihood in $\Delta_{Latest \text{visited}} \setminus \gamma$ is chosen, i.e., $c_i = \max_{\Delta_{Latest \text{visited}} \setminus \gamma} H[i]$. These three strategies enable our approach to select as much of the whole diagnosis as possible and to avoid redundant components. Hence, our approach is neither union-based search nor powerset-based search. We aim to avoid redundant components in a union-based search.

C. CONFIGURATION NEIGHBOR WITH ASPIRATION APPROACH USING THE SUB-DEGREE (CNASD)

We show the main flow of the CNASD algorithm in Figure 2. The key idea behind our algorithm is to start by obtaining the best diagnosis with the highest probability as the initial repair sequence. We then acquire all configuration neighbors of this sequence. From the configuration neighbors, the component with the highest probability is added into the repair sequence via the CNA strategy. However, there may be more than one such component. Then, the minimal sub-degree strategy will be used to find the component with the highest probability in the latest visited diagnosis with minimal sub-degree. The BDHC strategy will be used if no configuration neighbor exists.

We now present our CNASD algorithm in detail. The pseudocode of CNASD is given in Algorithm 1.
maxSteps is set as the maximum number of attempts on line 2 of Algorithm 1. A reasonable number of attempts should be set to avoid the addition of a large number of components into the repair sequence. These are useless for users because they cannot fix such a mass of components in the system. We check whether a configuration neighbor exists via the HasConfigurationNeighbors() method on line 3 of Algorithm 1.

1) STEP 1. (INITIALIZATION)
The CNASD algorithm starts by initializing the batch repair sequence. The BDHC strategy is used to find the best diagnosis with the highest probability component on line 1 of Algorithm 1. Algorithm 2 implements this procedure.

Note that the task of the method notAddToRepair() is to determine whether the current diagnosis has been added to the batch repair sequence on line 6 of Algorithm 2. A diagnosis that has appeared in the batch repair sequence is not selected again.

2) STEP 2. (FIND THE CONFIGURATION NEIGHBOR WITH ASPIRATION)
After the initialization process, we acquire all configuration neighbors of the current batch repair sequence as the search space. Then, the aspiration mechanism ensures that the highest-probability component is added into the batch repair sequence via the CNA strategy. This procedure is implemented in Algorithm 3. However, if no configuration
Algorithm 3 CNA: Configuration Neighbor With Aspiration

Input: $\gamma$: list of components, $DiagPs$: list of (diagnosis, probability), $HS$: list of (component, probability)

Output: a component

1 $HSsort = HS.sort();$ # sort HS by probability in descending order
2 $DiagPssort = DiagPs.sort();$ # sort DiagPs by probability in descending order
3 $CNeighbors = CN(\gamma, DiagPs);$ # find all configuration neighbors of $\gamma$
4 $CNeighborsA = \text{HighestProbability}(CNeighbors, HS);$ # find the components with highest probability in $CNeighbors$
5 if $CNeighborsA.size() = 1$ then
6 $\text{comp} = CNeighborsA.get(0);$ 
7 else
8 $\text{comp} = \text{MinSubdegree}(\gamma, CNeighborsA, DiagPssort, HSsort);$
9 return $\text{comp}$

Algorithm 4 MinSubdegree: Minimal Sub-Degree

Input: $\gamma$: list of components, $CNeighborsA$: list of components, $DiagPs$: list of (diagnosis, probability), $HS$: list of (component, probability)

Output: a component

1 forall the $\text{com in CNeighborsA}$ do
2 $\text{diags} = \text{diags}.add(\text{com}, DiagPs);$ # find the diagnosis including $\text{com}$
3 $\text{diags}_{-\text{subdegree}} = \text{diags.subdegree}(\gamma);$ # find the list of (diagnosis, sub-degree)
4 $\text{diags}_{-\text{subdegree}} = \text{diags}_{-\text{subdegree}}.sort();$ # sort $\text{diags}_{-\text{subdegree}}$ by sub-degree in ascending sort
5 if $\text{diags}_{-\text{subdegree}}$ has only one diagnosis with minimal sub-degree then
6 $\text{comp} = \text{HighestProbability}(\text{DifferenceSet(\text{diags}_{-\text{subdegree}}[0], \gamma), HS});$
7 else
8 $\text{comp} = \text{HighestProbability}(\text{DifferenceSet(\text{diags}_{-\text{subdegree}}.getLatestVisitedDiagnosis(\gamma), HS});$
9 return $\text{comp}$

neighbor exists, we use the BDHC strategy on line 8 of Algorithm 1.

3) STEP 3. (FIND THE COMPONENT IN THE LATEST DIAGNOSIS WITH THE MINIMAL SUB-DEGREE)

The aspiration mechanism may find more than one component with the highest probability. At this point, the minimal sub-degree strategy is useful. We find the diagnoses that include $\text{com}$ that were obtained in step 2 (Algorithm 3). When only one diagnosis has a minimal sub-degree, the highest-probability component is chosen, which is in the difference set of the current batch repair sequence and this diagnosis. Sometimes, more than one diagnosis has a minimal degree. The latest-visited diagnosis is taken into account. We select the highest-probability component from the difference set of the current batch repair sequence and then this diagnosis. This process is outlined in Algorithm 4.

In summary, we propose three strategies to obtain an optimal batch repair sequence. The initialization process uses the BDHC strategy. The CNA and minimal sub-degree strategies guide us in approaching the optimal batch repair sequence with the configuration neighbors (if they exist) as the search space. If no configuration neighbors exist, our approach follows the BDHC strategy.

D. ANALYSIS OF COMPUTATIONAL COMPLEXITY

For exploring the search space [1], [20], the powerset-based search considers a union of $k$ components in each search iteration process, where $k$ is a parameter. Union-based search obtains the union of $k$ diagnoses in each search iteration process, where $k$ is a parameter. A higher $k$ incurs a larger search space. Consequently, higher computational complexity results. However, a higher $k$ also increases the length of the batch repair sequence considered, and therefore, using a higher $k$ value can potentially obtain a better batch repair sequence than using lower $k$ values.

Note that our approach is neither a powerset-based search nor a union-based search. In our approach, the search space is a hybrid search space, namely, one that can choose an entire diagnosis or a single component. In our approach, the configuration neighbors become the current search space, if they exist, according to the CNA strategy. If configuration neighbors do not exist, the diagnoses not used for the batch repair sequence become the current search space. The BDHC strategy will work in this case. Note that in each search iteration process, the search space is a single kind of search space, i.e., it consists of either diagnoses or components not in the current sequence.

For example, take a system that has $m$ diagnoses and $n$ components. In a powerset-based search, the search space in each search iteration process can be as large as $C_n^k$. For a union-based search, the search space in each search iteration process can be as large as $C_n^k$. In our approach, the search space in each search iteration process can be as large as $C_n^1$ or $C_n^k$. The one extreme case of our approach is the case where no configuration neighbor exists in any iteration. Thus, the search space of our approach in each iteration can be as large as $C_n^1$. The other extreme case is that configuration neighbors exist in every iteration. Generally, the number of configuration neighbors is less than the number of system components. Specifically, the search space of our approach in each iteration is smaller than $C_n^1$. 

35922
E. RELATED WORKS

Local search is an efficient algorithm to solve the optimization problem. Local search has very good performance in many areas. Cai and Su [24] propose a new heuristic based on a checking strategy for a stochastic local search algorithm to solve the satisfiability problem (SAT). Cai et al. [25] propose a number of new ideas for local search for the partial maximum satisfiability problem. Wang et al. [23] propose a local search method to solve the minimum-weight dominating set problem based on two-level configuration checking. Song et al. [26] propose an iterated local search algorithm for solving university course timetabling problems. Saviniec and Constantino [27] propose iterated local search and variable neighborhood search-based algorithms for high school timetabling problems. Zhang et al. [5] propose a learning-based local search algorithm for the graph coloring problem. Michalak [29] propose a new local search method for the multiobjective firefighter problem to reduce the computational cost.

The key in local search is the definition of “neighbor” and the heuristic strategy. The abovementioned works on local search define “neighbor” and propose corresponding heuristic strategies in many areas. The advantages of local search are efficiency and flexibility in finding optimal solutions. This paper defines the “neighbor” in batch repair problems and selects heuristic strategies to guide search processes. In addition, local search, as a greedy algorithm, cannot be proven to obtain the optimal solution. Namely, the local search approach obtains the (approximate) optimal solution.

For the BRP, in this paper, our goal in the local search is to find the minimal cost of fixing the system. We incorporate the characteristics of a diagnosis problem, where the diagnosis is a reasonable explanation of the system failure. However, it is always impossible to repair the system by only repairing the components in a single diagnosis. Hence, to fix the system, repairing the components in the union of many diagnoses is a reasonable solution. To achieve minimal cost, it is wise to consider as few components as possible. This paper proposes three strategies to find the balance between incurring the minimal cost and fixing the system within the maximum number of iterations. Certainly, there is a case in which no repair sequence that fixes the system is found within the maximum number of iterations. In the experiment, we set a reasonable maximum number of iterations and show the percentage of instances solved. Our local search approach can solve most instances.

System fixed is the premise of BRP to get minimal cost. However, the simple cost-oriented ways in previous works neglect this premise. Actually, repairing the system earlier will minimize the repair cost.

Our algorithm start with initial batch repair sequence with BDHC. We avoid the null initial batch repair sequence in previous works. And the BDHC is used to ensure the system fixed first and next obtain minimal cost. Because we also consider the scores on diagnosis and component.

Second, CNA and sub-degree are used to update batch repair sequence. And they update sequence with a component for minimal cost first and system fixed second. BDHC is also used to update batch repair sequence. It update sequence with the components in a diagnosis for system fixed first and minimal cost second. These are because the diagnosis is the reasonable explanation of abnormal system. To add the diagnosis into repair sequence aims to fix system. Adding a component aims to minimal cost. And the CNA and sub-degree consider score on component. It also makes adding a component for fixing system to some extents.

Thence, our strategies aim to both system fixed and minimal cost. But different strategies consider different priorities on system fixed and minimal cost. Overall, our algorithm find the balance between incurring the minimal cost and fixing the system within the maximum number of iterations and the following experimental results also prove this.

Troubleshooting algorithms, in general, plan a sequence of actions that are intended to fix an abnormally behaving system. Troubleshooting algorithms have been applied to many areas. Heckerman et al. [16] propose a decision-theoretic approach for generating troubleshooting plans for automobile systems. Friedrich and Nejdl [17] propose a cost model for troubleshooting problems in power transmission network systems. Pernestål et al. [18] propose a decision-theoretic troubleshooting system that is developed to handle external interventions on automotive vehicle systems. Torta et al. [19] propose a novel abductive problem-solving method that extends previous work on measurement selection in model-based reasoning and on decision-theoretic troubleshooting by considering structured representations of the domain. However, all the above research on troubleshooting assume that components are repaired one at a time. Stern et al. [20] propose batch repair to solve troubleshooting problems and apply this approach it to circuit systems for the first time. This work shows the efficiency of the batch repair method. Then, Shnitzky et al. [1] propose heuristic search algorithms to solve batch repair in circuit systems.

F./APPLICABILITY AND LIMITATIONS

For the workflow as shown in Figure 1, our algorithm for the BRP requires corresponding diagnosis results, i.e., the health state, diagnosis and score. Therefore, our algorithm for the BRP works as long as it provides diagnostic results. In this paper, these results are obtained by the MBD algorithm. The experiment in our paper uses the ISCAS'85 and ITC'99 combination circuit systems, which are always used in the evaluation of MBD algorithms. The key is the model process, which is important for determining the acceptable input form of the diagnosis algorithm. The circuit system need to be modeled in a CNF or WCNF file. Then, the SAT- or MaxSAT-based diagnosis algorithm uses the SAT or MaxSAT solver to obtain diagnosis results.

In addition to the above actual combination circuit systems, general simulations are also an important source of circuit
TABLE 2. Average cost and maximum cost incurred when overhead = 15 and 25.

| Circuit | Overhead=15 | Overhead=25 |
|---------|-------------|-------------|
|         | A* | HC | B(2) | CNASD | A* | HC | B(2) | CNASD |
| c432    | 58 | 82 | 68 | 102 | 54 | 97 | 49 | 99 | 100 | 97 | 187 | 74 | 116 | 71 | 167 | 68 | 102 | 100 |
| c499    | 63 | 122 | 95 | 257 | 84 | 145 | 63 | 103 | 100 | 96 | 154 | 113 | 401 | 108 | 376 | 81 | 135 | 100 |
| c880    | 91 | 201 | 82 | 194 | 111 | 234 | 78 | 166 | 100 | 138 | 398 | 126 | 377 | 204 | 452 | 116 | 278 | 100 |
| c1355   | 94 | 233 | 103 | 185 | 124 | 254 | 93 | 186 | 99 | 144 | 326 | 227 | 378 | 242 | 401 | 132 | 258 | 99 |
| c1908   | 113 | 198 | 142 | 301 | 118 | 256 | 106 | 174 | 99 | 215 | 198 | 201 | 432 | 225 | 468 | 216 | 411 | 149 | 386 | 98 |
| c2670   | 107 | 261 | 110 | 351 | 123 | 312 | 99 | 215 | 99 | 223 | 417 | 274 | 399 | 283 | 472 | 209 | 385 | 99 |
| c3540   | 118 | 236 | 149 | 315 | 183 | 254 | 113 | 201 | 99 | 223 | 417 | 274 | 399 | 283 | 472 | 209 | 385 | 99 |
| b14     | 99 | 186 | 109 | 225 | 124 | 219 | 83 | 185 | 100 | 167 | 301 | 144 | 322 | 158 | 299 | 121 | 276 | 100 |
| b15     | 113 | 273 | 111 | 256 | 134 | 312 | 94 | 251 | 97 | 183 | 376 | 192 | 472 | 201 | 461 | 156 | 336 | 97 |
| b16     | 143 | 309 | 153 | 321 | 133 | 369 | 127 | 276 | 98 | 221 | 399 | 235 | 427 | 211 | 484 | 187 | 377 | 98 |
| b17     | 135 | 357 | 173 | 387 | 156 | 446 | 119 | 336 | 98 | 230 | 467 | 256 | 483 | 222 | 398 | 207 | 361 | 98 |
| b18     | 157 | 377 | 177 | 421 | 182 | 398 | 135 | 353 | 97 | 199 | 498 | 249 | 522 | 261 | 434 | 172 | 402 | 97 |
| b19     | 209 | 397 | 198 | 376 | 222 | 409 | 173 | 349 | 97 | 290 | 487 | 275 | 472 | 257 | 505 | 239 | 399 | 97 |
| b20     | 189 | 368 | 167 | 378 | 173 | 351 | 155 | 299 | 100 | 276 | 492 | 258 | 472 | 233 | 488 | 209 | 427 | 100 |
| b21     | 172 | 401 | 197 | 399 | 201 | 427 | 170 | 355 | 97 | 253 | 663 | 296 | 572 | 322 | 693 | 248 | 487 | 97 |
| b22     | 209 | 442 | 211 | 467 | 236 | 494 | 185 | 371 | 97 | 301 | 685 | 326 | 694 | 298 | 726 | 264 | 599 | 97 |

systems, such as field programmable gate arrays (FPGA). Due to their low power consumption and reconfigurability, FGPs have been successfully applied to hardware acceleration [41]–[44]. For a simulation circuit system based on FPGA, our algorithm for BRP will work as long as it can provide diagnostic results, i.e., the health state, diagnosis and score. This simulation process is different from the MBD process in this paper. The MBD process is based on the model. This process needs to model the system. However, the simulation process is based on data and the corresponding relationship formula. As long as it can provide the scores on diagnoses and the health state of components, our local search algorithm will work.

Because the repair process begins after the diagnosis problem arises, the key consideration is whether the diagnosis process can provide information on general simulations or real systems.

IV. EXPERIMENTAL EVALUATIONS

This section evaluates our CNASD approach to computing the incurred cost of the optimal batch repair action. We show the average cost incurred before the system is fixed in the experimental results.

A. EXPERIMENT PRELIMINARIES

• Implementation

Our approach was implemented in Java. Moreover, we adopted Direct-MS\textsubscript{SAT} [37] as the basic diagnosis method in our approach, which was a complete MBD approach using SAT. Actually, any completed diagnosis algorithm could be selected here. We selected only one complete MBD approach using SAT that performed well. More importantly, we obtained minimal-subset diagnoses in order of cardinality.

• Competitors

We compared our approach with the batch repair algorithms that use heuristic search, such as A* variant, HC and depth-limited breadth-first search [1]. In [1], \textit{costFFP} had the best performance. Hence, we selected this enhanced wasted cost utility function as the heuristic for the compared approaches. The depth-limited breadth-first search, denoted as B(n), limits the depth via n. In [1], B(2) had better performance than B(1) among the B(n) algorithms. Hence, we chose B(2) to represent the depth-limited breadth-first search.

• Benchmarks

Shinitzky et al. [1] performed experiments on a standard Boolean circuit benchmark including c432, c880 and 74182, which are from the 74XXX and ISCAS’85 circuit systems. As the circuit in 74XXX systems is small, we mainly performed experiments on ISCAS’85 systems and ITC’99 systems in this paper. The ISCAS’85 systems are standard combinational Boolean circuits, which are generally used as the basis for comparing results in the circuit diagnosis field. ITC’99 systems are standard combinational Boolean circuits, which are generally used in areas such as diagnosis, design for testability (DFT) and automatic test pattern generation (ATPG).

We ran instances using random observations. We injected 1 to 5 faults into the circuits. The prior probability of being faulty was set to 0.01 for each gate. For each observation, the subset-minimal diagnoses were found whose cardinality is less than or equal to the number of injected faults. We set \textit{maxSteps} as \(|\text{Comps}|/10.

• Experimental Setup

The experiments were performed on a machine running Windows 10, 64-bit, with an Intel i5-4590 3.30 GHz processor and 8 GB of physical memory.

B. EXPERIMENT RESULTS

Table 2 shows the average cost and maximum cost incurred with different overhead costs until the system is fixed. The rows in these tables show the results for the different
FIGURE 3. Runtime comparison with multiple actual faults on ISCAS'85 circuit system. (Ave. denotes average runtime and Max. denotes maximum runtime).
FIGURE 4. Runtime comparison with multiple actual faults on ITC’99 circuit system. (Ave. denotes average runtime and Max. denotes maximum runtime).

TABLE 3. Main acronyms and corresponding meaning in this paper.

| Acronym | Full name | Function |
|---------|-----------|----------|
| BRP [1] | batch repair problem | An efficient way to solve troubleshooting problem |
| CNASD   | local search approach for batch repair with configuration neighbor and sub-degree | The main algorithm proposed in this paper to solve BRP |
| BDHC    | best diagnosis with highest probability component | A strategy used in CNASD to obtain initial batch repair sequence and search a diagnosis for updating batch repair sequence |
| CNA     | configuration neighbor with aspiration | An efficient way to solve fault diagnosis problem |
| MBD [32], [33] | model based diagnosis | A strategy used in CNASD to search a component for updating batch repair sequence |
| CN      | configuration neighbor | A definition used in CNASD about the neighbor for batch repair sequence |
| HC [1]  | hill climbing | A heuristic method can be used for BRP |
| SAT [2] | satisfiability | A solver to solve problems expressed as logical formulas, such as MBD |
| MaxSAT [37] | maximum satisfiability | A solver to solve problems expressed as logical formulas, such as MBD |
| WCNF [2], [37] | (weighted) conjunctive normal form | A file form used for SAT/MaxSAT solver |

benchmark systems (on ISCAS’85 and ITC’99 circuit systems). The columns in these tables show the results for the different search algorithms, where A*, HC, B(2) and CNASD correspond to A* variant, hill climbing, breadth-first search with a depth bound of two, and our heuristic search algorithm CNASD, respectively. The results in Table 2 are given for the overhead costs of 15 and 25, and the best result in each case is bolded. Column A. denotes the average cost, and column M. denotes the maximum cost. In CNASD, column S. (%) denotes the percentage of instances solved.

As can be observed, our CNASD approach exhibits outstanding performance and can solve most instances. This result is relatively consistent over these two cases with different overhead costs. Obviously, the cost of batch repair increases as the repair overhead increases. As expected, as the scale of the system increases, the cost of batch
repair increases. To fix the system, our approach considers the reasonability of a diagnosis (whether the diagnosis can explain the system’s current abnormal behavior) and avoids redundant components in diagnoses as much as possible. Hence, our approach yields excellent results. Whether for the average cost or maximum cost, our algorithm performs the best in most cases. On average, our algorithm obtains better repair sequences. As can be observed from the maximum cost, our algorithm performs well and has better stability on repair costs.

In addition, our algorithm efficiently reduces the search space. The comparisons of average runtime and maximum runtime on ISCAS’85 and ITC’99 circuit systems are shown in Figure 3 and 4. As can be observed from Figure 3 and 4, compared with A*, HC and B(2), the CNASD algorithm has a large increase in average runtime on ISCAS’85 and ITC’99 circuit systems. Additionally, as the number of faulty components increases, the increase becomes more significant, up to an order of magnitude greater. This result is due to the reduction of the search space by the CNASD algorithm. Therefore, our algorithm exhibits a reduced runtime. Comparing the average runtime and maximum runtime of the algorithms, we find that the CNASD algorithm has a smaller gap. The CNASD algorithm has a more stable runtime.

V. CONCLUSION

This paper presented a novel local search approach to solving the BRP with configuration neighbors and sub-degree. This method adopts three strategies, namely, BDHC, CNA and minimal sub-degree. First, the BDHC strategy is used to initialize the batch repair sequence. Then, the configuration neighbor is used to limit the search space. If no configuration neighbor exists, the BDHC strategy continues to play a role. Second, the CNA and minimal sub-degree strategies are used to find the next candidate component to add to the batch repair sequence. The experimental evaluation used the ISCAS’85 and ITC’99 benchmarks. We compared the average cost, average runtime, maximum cost and maximum runtime incurred by our method with those incurred by the methods of A* variant, HC and depth-limited breadth-first search [1] in state-of-the-art batch repair algorithms with heuristic search. The results are unequivocal; our approach outperforms the other approaches in terms of average cost and average runtime incurred until the system is fixed. Moreover, our approach has better stability on cost and runtime.

In the future, we would like to seek further optimization using the CNASD approach. In this paper, we have neglected an important and commonly available source of information about a circuit system, namely, its topology. The topology may help us to learn the relationships among diagnoses and obtain a more efficient strategy. In addition, we want to apply this approach to other areas, such as program and spreadsheet systems. Many diagnosis algorithms for those systems exist. Thus, the key task is to model the configuration neighbor for those systems and to find an efficient search strategy.

APPENDIX

We present a list of acronyms with their meanings in Table 3.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their valuable comments.

REFERENCES

[1] H. Shinitzky, R. T. Stern, and M. Kalech, “Batch repair with heuristic search,” in Proc. 9th Annua. Symp. Combinat. Search, New York, NY, USA, 2016, pp. 141–142.
[2] A. Metodi, R. Stern, M. Kalech, and M. Codish, “A novel SAT-based approach to model based diagnosis,” J. Artif. Intell. Res., vol. 51, no. 1, pp. 377–411, Jan. 2014.
[3] Y. Chen, S. Safarpour, A. Veneris, and J. Marques-Silva, “Spatial and temporal design debug using partial MaxSAT,” in Proc. 19th ACM Great Lakes Symp. VLSI (GLSVLSI), Boston, MA, USA, 2009, pp. 345–350.
[4] J. Marques-Silva, M. Janota, A. Ignatiev, and A. Morgado, “Efficient model based diagnosis with maximum satisfiability,” in Proc. 24th Int. Joint Conf. Artif. Intell., Buenos Aires, Argentina, 2015, pp. 1966–1972.
[5] J. Zhang, F. Ma, and Z. Zhang, “Faulty interaction identification via constraint solving and optimization,” in Proc. 15th Int. Conf. Theory Appl. Satisfiability Test. (SAT), Trento, Italy, 2012, pp. 186–199.
[6] A. R. Ramos, O. Llanes-Santiago, J. M. B. de Lázaro, C. C. Corona, A. I. S. Neto, and J. V. Gallego, “A novel fault diagnosis scheme applying fuzzy clustering algorithms,” Appl. Soft Comput., vol. 58, pp. 605–619, Sep. 2017.
[7] N. M. Nor, M. A. Hussain, and C. R. C. Hassan, “Fault diagnosis and classification framework using multi-scale classification based on kernel Fisher discriminant analysis for chemical process system,” Appl. Soft Comput., vol. 61, pp. 959–972, Dec. 2017.
[8] C. H. Lo, Y. K. Wong, and A. B. Rad, “Bond graph based Bayesian network for fault diagnosis,” Appl. Soft Comput., vol. 11, no. 1, pp. 1208–1212, Jan. 2011.
[9] M. Liu, D. Ouyang, and L. Zhang, “A novel approach for improving quality of health state with difference degree in circuit diagnosis,” Int. J. Speech Technol., vol. 48, no. 11, pp. 4371–4381, Jun. 2018.
[10] B. C. Williams and P. P. Nayak, “A model-based approach to reactive self-configuring systems,” in Proc. 13th Nat. Conf. Artif. Intell., 8th Innov. Appl. Artif. Intell. Conf., Portland, OR, USA, 1996, pp. 971–978.
[11] A. Feldman, H. V. de Castro, A. van Gemund, and G. Provan, “Model-based diagnostic decision-support system for satellites,” in Proc. IEEE Aerosp. Conf., Big Sky, MT, USA, Mar. 2013, pp. 1–14.
[12] P. Struss and C. J. Price, “Model-based systems in the automotive industry,” AI Mag., vol. 24, no. 4, pp. 17–34, Apr. 2004.
[13] D. Janneck and T. Schmitz, “Model-based diagnosis of spreadsheet programs: A constraint-based debugging approach,” Automat. Softw. Eng., vol. 23, no. 1, pp. 105–144, Feb. 2014.
[14] S. M. De Oca and V. Puig, “Fault diagnosis of nonlinear systems using LPV interval observers,” in Proc. 20th Int. Workshop Princ. Diagnosis, Stockholm, Sweden, 2009, pp. 14–17.
[15] M. Buciakowski, M. Witzczak, V. Puig, D. Ronotto, F. Nejjarl, and J. Korbicz, “A bounded-error approach to simultaneous state and actuator fault estimation for a class of nonlinear systems,” J. Process Control, vol. 52, pp. 14–25, Apr. 2017.
[16] D. Hecker, J. S. Breese, and K. Rommelse, “Decision-theoretic troubleshooting,” Commun. ACM, vol. 38, no. 3, pp. 49–57, Mar. 1995.
[17] G. Friedrich and W. Nejdl, “Choosing observations and actions in model-based diagnosis/repair systems,” in Proc. 3rd Int. Conf. Princ. Knowl. Represent. Reasoning, Cambridge, MA, USA, 1992, pp. 489–498.
[18] A. Permešl, M. Nyberg, and H. Warning, “Modeling and inference for troubleshooting with interventions applied to a heavy truck auxiliary braking system,” Eng. Appl. Artif. Intell., vol. 25, no. 4, pp. 705–719, Jun. 2012.
[19] G. Torta, L. Anselma, and D. T. Dupré, “Exploiting abstractions in cost-sensitive abductive problem solving with observations and actions,” AI Commun., vol. 27, no. 3, pp. 245–262, Mar. 2014.
[20] R. Stern, M. Kalech, and H. Shinitzky, “Implementing troubleshooting with batch repair,” in Proc. 30th AAAI Conf. Artif. Intell., Phoenix, AZ, USA, 2016, pp. 769–775.
[21] T. Zamir, R. T. Stern, and M. Kalech, “Using model-based diagnosis to improve software testing,” in *Proc. 28th AAAI Conf. Artif. Intell.*, Quebec City, QC, USA, 2014, pp. 1135–1141.

[22] R. Stern, M. Kalech, S. Rogov, and A. Feldman, “How many diagnoses do we need?” *Artif. Intell.*, vol. 248, pp. 26–45, Jul. 2017.

[23] Y. Wang, S. Cai, and M. Yin, “Local search for minimum weight dominating set with two-level configuration checking and frequency based scoring function,” *J. Artif. Intell. Res.*, vol. 58, pp. 267–295, Feb. 2017.

[24] S. Cai and K. Su, “Configuration checking with aspiration in local search for SAT,” in *Proc. 26th AAAI Conf. Artif. Intell.*, Toronto, ON, Canada, 2012, pp. 434–440.

[25] S. Cai, C. Luo, J. Lin, and K. Su, “New local search methods for partial MaxSAT,” *Artif. Intell.*, vol. 240, pp. 1–18, Nov. 2016.

[26] T. Song, S. Liu, X. Tang, X. Peng, and M. Chen, “An iterated local search algorithm for the university course timetabling problem,” *Appl. Soft Comput.*, vol. 68, pp. 597–608, Jul. 2018.

[27] L. Savinieco and A. A. Constantino, “Effective local search algorithms for high school timetabling problems,” *Appl. Soft Comput.*, vol. 60, pp. 363–373, Nov. 2017.

[28] Y. Zhou, B. Duval, and J.-K. Hao, “Improving probability learning based local search for graph coloring,” *Appl. Soft Comput.*, vol. 65, pp. 542–553, Apr. 2018.

[29] K. Michalak, “ED-LS—A heuristic local search for the multiobjective firefighter problem,” *Appl. Soft Comput.*, vol. 59, pp. 389–404, Jan. 2017.

[30] R. Li, H. Liu, X. Wu, J. Wu, and M. Yin, “An efficient local search algorithm for the minimum k-dominating set problem,” *IEEE Access*, vol. 6, pp. 62062–62075, 2018.

[31] Y. Zhou, C. Qu, Y. Wang, M. Fan, and M. Yin, “An improved heuristic algorithm for the partial vertex cover problem,” *IEEE Access*, vol. 7, pp. 17389–17402, 2019.

[32] R. Reiter, “A theory of diagnosis from first principles,” *Artif. Intell.*, vol. 32, no. 1, pp. 97–130, Apr. 1987.

[33] J. de Kleer and B. C. Williams, “Diagnosing multiple faults,” in *Proc. 26th AAAI Conf. Artif. Intell.*, Toronto, ON, Canada, 2012, pp. 793–799.

[34] A. Metodi, R. Stern, M. Kalech, and M. Codish, “Compiling model-based diagnosis to Boolean satisfaction,” in *Proc. 26th AAAI Conf. Artif. Intell.*, Toronto, ON, Canada, 2012, pp. 793–799.

[35] A. Roychoudhury and S. Chandra, “Formula-based software debugging,” *Commun. ACM*, vol. 59, no. 7, pp. 68–77, Jun. 2016.

[36] M. Liu, D. Ouyang, S. Cai, and L. Zhang, “An improved memetic algorithm for SAT,” in *Proc. 26th AAAI Conf. Artif. Intell.*, Toronto, ON, Canada, 2012, pp. 793–799.

[37] M. Li, D. Ouyang, S. Cai, and M. Yang, “Optimising FPGA-based neural network accelerators using fast Winograd algorithm,” in *Proc. Int. Conf. Hardw./Softw. Codexen Systems. (CODES+ISSS)*, Turin, Italy, Sep. 2018, pp. 1–2.

[38] C. Wang, L. Gong, Q. Yu, X. Li, Y. Xie, and X. Zhou, “DLAU: A scalable deep learning accelerator unit on FPGA,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 36, no. 3, pp. 513–517, Mar. 2017.

[39] C. Wang, X. Li, Y. Chen, Y. Zhang, O. Diessel, and X. Zhou, “Service-oriented architecture on FPGA-based MPPSoC,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 10, pp. 2993–3006, Oct. 2017.

MENG LIU received the bachelor’s degree from Jilin University, in 2015, where she is currently pursuing the Ph.D. degree with the College of Computer Science and Technology. Her research interests include model-based diagnosis, local search, batch repair, and satisfiability (SAT).

DANTONG OUYANG received the M.S. degree and the Ph.D. degree in computer science from Jilin University, in 1993 and 1998, respectively. She is currently a Professor with the College of Computer Science and Technology, Jilin University. Her current research interests include model-based diagnosis and automated reasoning.

NAIYU TIAN received the bachelor’s degree from Jilin University, in 2016, where she is currently pursuing the Ph.D. degree with the College of Computer Science and Technology. Her research interests include model-based diagnosis, satisfiability (SAT), and #SAT.

HUA LI received the M.S. degree in thoracic surgery from The First Hospital of Jilin University, Changchun, China, in 2009. She is currently an Attending Doctor with the Department of Abdominal Ultrasound, The First Hospital of Jilin University. Her main research interests are model-based diagnosis, patient stratification, disease prediction, and diagnosis.