International Journal of Gender, Science and Technology
..ISSN: 2040-0748

UGC Care Group I Journal
Vol-10 Issue-02 Aug 2021

Utilization of Reinforcement Learning for Automatic Generation of Reconfiguration Blueprints in Integrated Modular Avionics (IMA) Systems.

• Dr. Subrat Kumar Mohanty,

Department of Electronics and Telecommunication, College of Engineering Bhubaneswar

***Abstract*— Enhancing the fault-tolerant capability of the integrated modular aircraft (IMA) system requires reconfiguration as a crucial method. The quality of the blueprints greatly affects operational performance after system reconfiguration. Consequently, reinforcement learning (RL) is added to the IMA system through the intelligent reconfiguration technique described in this letter. The proposed method combines the key elements of the RL model with the well-established IMA reconfiguration model to produce high-quality and cost-effective designs. The results of the experiment indicate that the proposed method can quickly generate feasible blueprints and also optimize load balancing and reconfiguration costs to enhance system performance.**

I. Introduction

INTEGRATED modular avionics (IMA) is one of the most potentialavionicssystemspresently.Withtheincreasing numberofIMAapplications,thesystemstructurebecomes highlycomplex.Thefailureofonecomponentwillinflu-enceothermodulesofthesystem.Nowadays,academicand industryadoptthedynamicallyreconfigurabletechnologyto guarantee the high reliability of IMA. Reconfiguration refers towhenoperationscenariochangesorasystemcomponent fails,thesystemreallocatessoftwareapplicationsontohard-waremodulesaccordingtothereconfigurationblueprint[1]. Thus,thesystemisrecoveredinashorttime.

The traditional reconfiguration methods usually train faults in advance and then load the training results onto the air-craft. However, if new untrained failure modes occur during flight, traditional methods cannot generate feasible reconfig-urationblueprintsrapidly.Tosolvetheaboveproblems,wedevise an intelligent reconfiguration algorithm, which learns repeatedly and accumulates experience in the course of state transfer. Note that the proposed algorithm is only appropriate for noncritical applications. For critical applications, fault tol-erance will be implemented through redundancy to guarantee the safety of the aircraft. Besides, software redundancy is fea-sible for fault tolerance [2]. However, if software redundancy isusedfornoncriticalapplicationsaswell,itwillincrease the complexity and resource demand of the system. The main contributions of our proposed approach are as follows.

1) Anintelligentreconfigurationalgorithmisproposed bycombiningthekeyfactorsofreinforcement learning (RL) with the established IMA recon-figuration model to generate feasible blueprintsinstantly.
2) The proposed method optimizes the costs required duringreconfigurationandtheloadbalanceof IMA after reconfiguration to ensure the quality of blueprints.

This letter is organized as follows. Section II provides related works for IMA reconfiguration. Section III intro-duces our proposed intelligent reconfiguration method for the IMA system. Section IV discusses the experimental setup and results. Finally, Section V concludes this letter.

II. RelatedWorks

Recent research proposes several reconfiguration schemes for IMA architectures. Cui *et al.* [3] suggested a decentralized reconfiguration technique, applying a concept called back-wards reconfiguration. Wang *et al.* [4] proposed a system reliabilitymodelingandverificationmethodbasedon finite-statemachinetheoryandsystemfailurelogicfor IMA.Pourmohseni*etal.*[5]presentedadeterminis-tic mapping reconfiguration methodology to enable pre-dictablereconfigurationsamongagivensetofmappings. Da Fontoura *et al.* [6] proposed a distributed reconfigurable architecture, which was applied in an avionic system case study to deal with fault management. Annighöfer *et al.* [7] presented an automated device type selection, sizing, and mapping method based on binary programming to improvetheperformanceofsystemarchitectures.Inconclusion, the current research on reconfiguration mostly focuses on systemmodelingandverification,andlimitedattentionhasbeenpai dtothegenerationmethodofblueprints. Hence, this letter mainly studies to obtain blueprints intelligently.

III. Intelligent Reconfiguration for IMA System

We initially established the system model according to the composition and characteristics of IMA to effectively describe the algorithm. Then, we design the corresponding resource constraints of reconfiguration and evaluation indicators of blueprints to quantify the validity and quality of the blueprint. Finally, we design the five critical elements of the RL model to complete the intelligent reconfiguration algorithm. The following are each step presented in detail.

A. IMA System Model

IMA can be simplified to a centralized hardware structure. An equipment cabinet contains multiple processor modules, and several partitions are set in a processor. One or more software applications are deployed in each partition which provides hardware resources for them. Different processors connect each other with the communication bus. System resources in a partition are shared by software applications deployed on it. Hardware resources are abstractly transformed into logic modules in IMA based on the idea of resource sharing. Hence, application M, partition P, and processor C models are established successively in combination with hardware and software requirements.

B. Resource Constraints and Evaluation Indicators

First, we design resource constraints to screen out effective blueprints. In terms of time, each partition of the IMA system has a fixed start time and execution time within a mainframe of the processor. All applications in one partition must be accomplished in the required time. In terms of memory, the sum of the memory occupied by applications must not exceed the available memory size of the partition.

Next, we design the evaluation indicators to pick out high-ability blueprints from these valid blueprints. We select load balance and reconfiguration costs to evaluate the pros and cons of reconfiguration blueprints. Reconfiguration costs are further subdivided into reconfiguration influence, recovery time, and reconfiguration degradation three indicators. The definitions of these four indicators are explained as follows.

1) Load Balance: Load balance can improve the execution speed of the application. Standard deviation is utilized to measure the discrete degree of the load in each partition. Thus, LB is calculated as

$$LB = 1 - 2\sqrt{\frac{1}{n_P}\sum_{j=1}^{n_P}\left(L_{P_j} - \bar{L}\right)^2} \tag{1}$$

where $\bar{L}$ represents the average resource load of all processors. L_{P_j} denotes the load vector on the partition P_j, as defined in

L_{P_j} ... use ... use

2) Reconfiguration Influence: Reconfiguration influence represents the impact of system reconfiguration on applications. It mainly measures the ratio of successful reconfiguration of applications. We classify the importance of applications into five grades 1–5. The larger the number is, the more significant the application is. Hence, In is calculated as

$$\mathrm{In} = \sum_{i=1}^{n_M} G_{M_i} \Big/ \sum_{i=1}^{N_M} G_{M_i} \tag{3}$$

where n_M represents the number of applications which finish reconfiguration successfully, and N_M denotes the sum of applications that requires reconfiguration in the failed processor. G_{M_i} is the important degree of application M_i.

3) Recovery Time: The recovery of applications seizes major time in system reconfiguration. To facilitate the analysis, the transmission time is ignored in this letter. We assume when multiple applications are deployed on the same processor, the reloading is serial and the recovery time is supposed to be calculated cumulatively; when multiple applications are deployed on different processors, the reloading is parallel and the recovery time is considered as the maximum reloading time among all processors. Hence, T_{re} is expressed as

$$T_{re} = 1 - \mathrm{Max}\, T_{C_k} / T_{\max} \tag{4}$$

where $T_{\max}$ represents the maximum reconfiguration time prespecified. T_{C_k} denotes the recovery time of the processor C_k, as defined in

$$T_{C_k} = \sum_{i=1}^{N_{re}} T_{M_i} \tag{5}$$

where N_{re} represents the number of applications reload in the processor C_k. T_{M_i} is the reconfiguration time of the application M_i, depending on the application size.

4) Reconfiguration Degradation: Degradation represents low-priority functions that are sacrificed to restore high-priority functions when redundant resources of the system are insufficient. Reconfiguration degradation primarily measures the functional completeness of the system after reconfiguration. Then, De is constructed as

$$De = 1 - \sum_{i=1}^{n_M^f} G_{M_i} \Big/ \sum_{i=1}^{n_M} G_{M_i} \tag{6}$$

where n_M^f represents the sum of applications which fail to reconfigure, and n_M denotes the total number of applications in the system. G_{M_i} is the important degree of application M_i.

C. Intelligent Reconfiguration Algorithm

This letter employs the Q-learning algorithm based on value function to find the optimal "action-selection" strategy. Hence,

$= \mu_1 \times C^{Pj}_{use} + \mu_2 \times M^{Pj}_{use}$

wedesigncriticalelementsofRLfromthefollowingaspects.

1) State: Statemeanstheconfigurationinformationof hardwareandsoftwareintheIMAsystem.Wedesignthecon-figurationblueprintsintheprocessofsystemreconfigura tion

$$s = <C_s, P_s, M_s, \text{Bind}> \quad (7)$$

where $C_s = \{C_1, C_2, \ldots, C_n\}$, $P_s = \{P_1, P_2, \ldots, P_m\}$, and $M_s = \{M_1, M_2, \ldots, M_l\}$ represent the set of processors, partitions, and applications, respectively. Bind $= \{M_i \rightarrow P_j \rightarrow C_k\}$ isthemappinginformationofapplicationsonpartitionsand processors.

2) Action: Actionmeansreconfiguringanapplicationon an available partition. Action a is defined as

$$a = re < M_i, P_j, C_k > \quad (8)$$

where re denotes the redeployment of the application $M_i$tothe partition $P_j$in the processor C_k. The transition function from old state to new state after choosing an action is defined in

$$s_i \xrightarrow{a} s_{i+1}. \quad (9)$$

3) Policy: Policy refers to the foundation for choosing an action from action space in a configuration state. We choose the ε-greedy policy, which accurately balances explorationand exploitation, as the intelligent reconfiguration policy of IMA. This policy selects an unknown action randomly with theprobabilityofasmallpositivenumber ε andselectsthe action that has the highest value in the current state with the probabilityof $1 - \varepsilon$.

4) RewardFunction: Rewardfunctionisutilizedforthe real-time evaluation of the new configuration blueprint afteran action is taken. If the current configuration state s changes toa new state s^rafter taking the application reconfiguration action$a$,thenwemustoffer$a$ascalarrewardsignalwhich comesfromtheevaluationfors^r. Thisletterdividesthereward intothreesituationsaccordingtothestates^r: 1Ifs^rdoes not satisfy the resource constraints, we give a negative reward to the error configuration; 2If s^rsatisfies the resource constraints,whereassomeapplicationsarestilldeployedonthe failed processor, the current configuration must perform the reconfiguration of applications to generate an effective state. Hence,wegivenorewardtothefaultconfiguration; 3If s^rsatisfies the resource constraints and no applications are deployedonthefailedprocessor,wegiveapositivereward assessed by evaluation indicators to the valid configuration. Therefore, the reward function is defined as

$$r(s,a) = \begin{cases} -1, & \text{errorconfiguration} \\ 0, & \text{faultconfiguration} \\ \eta_1 \times De + \eta_2 \times In + \eta_3 \times T_{re} + \eta_4 \times LB & \end{cases} \quad (10)$$

TABLEI
ATTRIBUTEDATAOFSOFTWAREAPPLICATIONS

Application Id	Deadline (ms)	Worst Case Execute Time(ms)	Memory (kb)	Important Degree
M_1	40	4	90	4
M_2	40	4	50	4
M_3	40	4	30	3
M_4	40	2	40	3
M_5	40	2	70	3
M_6	40	4	90	2
M_7	20	4	80	2
M_8	40	4	60	3
M_9	20	4	50	3
M_{10}	40	4	95	3

TABLEII
ATTRIBUTEDATAANDINITIALCONFIGURATIONINFORMATIONOFIMA

Hardware Resources	MF (ms)	Partition	Memory (kb)	Execution Time(ms)	Process
C_1	40	P_1	256	10	/
		P_2	128	10	M_2
		P_3	128	20	M_1
C_2	40	P_1	512	40	M_3 M_4 M_5 M_6
C_3	40	P_1	256	20	M_7
		P_2	256	20	M_8
C_4	40	P_1	128	16	M_9
		P_2	128	8	/
		P_3	256	16	M_{10}

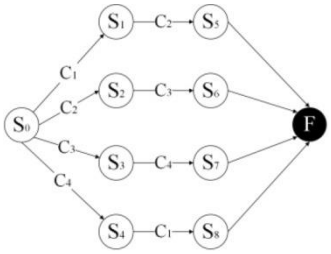

Fig.1.Statemigrationforintelligentreconfiguration.

state attained by executing these actions is the optimal reconfiguration blueprint of IMA. The iterative formula of Q value is defined as

$$\text{New}Q(s,a) = Q(s,a) + \alpha r(s,a) + \gamma \max Q^r(s^r, a^r) - Q(s,a) \quad (11)$$

vaildconfiguration

where $r(s,a)$ represents the reward for choosing reconfigurationaction$a$insystemconfigurations.De,In,T_{re},and LB are reconfiguration degradation indicator, reconfiguration influence indicator, recovery time indicator, and load balancingindicator,respectively.Theweightsofthesefourindicators aredenotedasη_1,η_2,η_3,andη_4,respectively.Inthisletter, $\eta_1=\eta_2=0.35$,$\eta_3=0.20$,and$\eta_4=0.10$.

5) ValueFunction: Valuefunctionchoosesthebestaction inacertainconfigurationbyevaluatingthe$Q$valueof each action taken. The Q value readjusts continuously, and eventually,thisvaluebecomesconvergentforaseriesofreconfigurationactionsinaconfigurationstate.Theconfiguration

where New$Q(s,a)$ represents the Q value acquired by takingaction a in system configuration s. $Q(s,a)$is the current$Q$value.$r(s,a)$ istherewardfunction.α andγrepresent the learning rate and reward decay, respectively. In this letter, $\alpha=0.01$and$\gamma=0.9$.max$Q^r(s^r,a^r)$is the maximum reward obtained through taking action a^ramong all possible actionsin the new reconfiguration state s^r.

IV. ExperimentalSetupandResults

The experiment aims to analyze the capacity of the intelligent reconfiguration algorithm to generate high-quality blueprints.First,wegatherthebasicconfigurationinformation oftheIMAsystemastheinputdataoftheexperiment,as

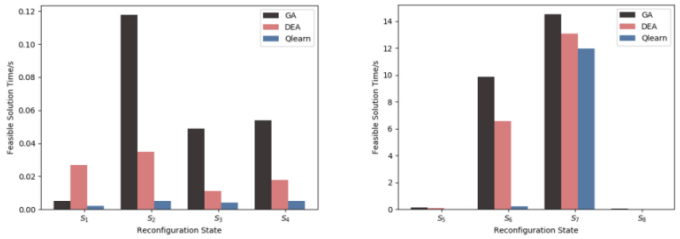

Fig.2.Comparisonoffeasiblesolutiontime.

TABLEIII
PartialRewardChangingofThreeAlgorithmsinS_6

Iteration Times	GA Reward	DEA Reward	Q-learning Reward
1	0.557051	0.536774	0.405899
10	0.557052	0.615153	0.430552
30	0.683252	0.632140	**0.902688**
…	…	…	…
100	0.557051	0.673471	0.717878
200	0.735766	0.683252	0.874006
300	0.683252	0.700755	0.743988
…	…	…	…
500	0.557051	0.742543	0.704668
600	0.777967	0.762086	0.714585
700	0.684917	0.784413	0.836358
…	…	…	…
1325	0.767004	0.809153	**0.918669**
2000	0.702601	0.869057	0.852338
3000	0.751248	0.90821	0.852337
…	…	…	…
3223	0.782748	**0.918669**	0.918669
11674	**0.918669**	0.918669	0.852337
14000	0.875483	0.918669	0.918669

represented in Tables I and II. From the initial configuration state, the experiment injects different processor faults to producediversereconfigurationstates.Fig.1depictsthetransitionoftheadoptedreconfigurationstates.Eachnoderepresentsthe reconfiguration state after the system failure.

We compare the time of obtaining feasible reconfiguration blueprints through the genetic algorithm (GA), the differen- tial evolution algorithm (DEA), and the proposed algorithm, respectively. These two comparison algorithms are frequently used to solve multiobjective optimization problems in recent years [8], [9].Fig.2showsthefeasiblesolutiontimeof three algorithms in eight reconfiguration states. It takes less time for the proposed algorithm to find feasible blueprintsthantheothertwoalgorithms.Takereconfigurationstate S_6,

for example, Table III shows the partial reward changing of threealgorithms.ThebolddatainTableIIIindicatethat theproposedalgorithmcanobtainabetterfeasiblesolution in fewer iteration times. In addition, the proposed algorithm can get the optimal reconfiguration blueprint in the minimum iteration times compared with the other two algorithms. Experimental results indicate that the intelligent reconfigura- tion algorithm can generate high-quality blueprints effectively based on the accumulated experience, thus guaranteeing the stability and security of the IMA system operation.

V. Conclusion

This letter includes the dynamic reconfiguration of the IMA system using the R Language program. In the design of intelligent configuration, we concentrate on the processes of system modeling, resource restrictions, evaluation indicators, and blueprint production.When obtaining the reconfiguration blueprints, the intelligent reconfiguration algorithm takes load balance and reconfiguration costs into account. This letter's research project not only offers a secure, workable solution to the IMA system's failure during reconfiguration, but it also successfully improves the system's fault tolerance and stability.

References

[1] P.Fu,S.H.Wang,andB.Liu,"Anoriginalapproachtoconstruct-ing test model for IMA blueprints," in *Proc. 2nd Int. Conf. Rel. Syst.Eng. (ICRSE)*, 2017, pp. 1–6.

[2] A. Girault, H. Kalla, M. Sighireanu, and Y. Sorel, "An algorithm forautomaticallyobtainingdistributedandfault-tolerantstaticschedules,"in*Proc. Int. Conf. Dependable Syst. Netw.*, 2003, pp. 159–168.

[3] Y.Q.Cui,J.Y.Shi,andZ.L.Wang,"Backwardreconfigurationmanage-mentformodularavionicreconfigurablesystems,"*IEEESyst.J.*,vol.12, no. 1, pp. 137–148, Mar. 2018.

[4] R.P.Wang,W.T.Lu,C.H.Zeng,X.Wang,Y.Z.Zhou,and C. Y. Liu, "Reliability modeling and verification method for dynamicreconfiguration system," in *Proc. Prognostics Syst. Health Manag. Conf.(PHM-Chongqing)*, 2018, pp. 941–947.

[5] B. Pourmohseni, S. Wildermann, M. Glaß, and J. Teich, "Hard real-timeapplicationmappingreconfigurationforNoC-basedmany-coresystems,"*Real-Time Syst.*, vol. 55, no. 2, pp. 433–469, 2019.

[6] A.A.daFontoura,F.A.M.doNascimento,S.Nadjm-Tehrani,and E. P. de Freitas, "Timing assurance of avionic reconfiguration schemesusingformalanalysis,"*IEEETrans.Aerosp.Electron.Syst.*,vol. 56,no.1, pp.95–106,Feb.2020.

[7] B.Annighöfer,E.Kleemann,andF.Thielecke,"Automatedselec-tion,sizing,andmappingofintegratedmodularavionicsmodules,"in*Pro c.IEEE/AIAA32ndDigit.AvionicsSyst.Conf.(DASC)*,2013, pp.2E2-1–2E2-15.

[8] M.Montazeri,R.Kiani,andS.S.Rastkhadiv,"Anewapproachtothe restart genetic algorithm to solve zero-one knapsack problem," in*Proc.IEEE4thInt.Conf.Knowl.BasedEng.Innovat.(KBEI)*,2017, pp.0050–0053.

[9] Y. Cheng, H. Wang, Z. Weng, J. Fang, and Z. Weng, "Optimization offlowshopschedulingcontrolstrategybasedonimproveddifferentiale vo-lution algorithm," in *Proc. 33rd Youth Acad. Annu. Conf. Chin. Assoc.Autom. (YAC)*, 2018, pp. 43–46.