

串并联系统中支持实时替换的混合冗余策略优化*

何盼, 郑志浩, 袁月, 谭春



(中国科学院 重庆绿色智能技术研究院, 重庆 400714)

通信作者: 何盼, E-mail: hepan@cigit.ac.cn

摘要: 在需要长时间可靠运行的软件系统中, 由于持续运行时间和任务响应速度的要求增加, 工作组件在被探测到失效后将被冗余组件实时替换. 但现有可靠性优化研究通常假设冷备份冗余在所有积极冗余组件失效后才使用. 针对支持实时替换的混合冗余策略, 对其冗余度优化分配进行研究. 该策略不仅能够保障系统可靠性, 而且能够保障系统性能, 故选用实时可用性和任务完成效率两类约束条件, 建立冗余配置代价最小化模型. 基于马尔可夫链理论对可靠性及性能两类系统指标进行定量分析; 采用数值计算方法对非线性的状态分析模型进行计算; 改进二元组编码遗传算法对上述优化问题进行求解. 采用实例对串并联系统中实时可用性及任务完成效率的分析进行了说明, 并对优化冗余分配模型进行了验证. 实验结果表明, 在相同冗余度下, 支持实时替换的混合冗余策略在任务完成效率方面优于传统的混合冗余策略. 所以, 在相同约束条件下不同混合冗余策略需要采用不同的冗余优化配置方案.

关键词: 冗余分配; 可靠性优化; 混合冗余策略; 遗传算法; 串并联系统

中图法分类号: TP316

中文引用格式: 何盼, 郑志浩, 袁月, 谭春. 串并联系统中支持实时替换的混合冗余策略优化. 软件学报, 2017, 28(2): 443-456. <http://www.jos.org.cn/1000-9825/5031.htm>

英文引用格式: He P, Zheng ZH, Yuan Y, Tan C. Optimization of mixed redundancy strategy with instant switching for series-parallel systems. Ruan Jian Xue Bao/Journal of Software, 2017, 28(2): 443-456 (in Chinese). <http://www.jos.org.cn/1000-9825/5031.htm>

Optimization of Mixed Redundancy Strategy with Instant Switching for Series-Parallel Systems

HE Pan, ZHENG Zhi-Hao, YUAN Yue, TAN Chun

(Chongqing Institute of Green and Intelligent Technology, The Chinese Academy of Sciences, Chongqing 400714, China)

Abstract: In long-time running reliable software systems, as the demand for continuous execution time and task response speed increases, the redundant component needs to be instantly switched when failure occurs. However, reliability optimization is often conducted under the assumption that cold standby redundancy is only activated when all active components fail. This paper tackles the redundancy allocation problem for a mixed redundancy strategy with instant switching to ensure system reliability as well as performance. The redundancy allocation model is built to minimize redundancy configuration cost under the transient availability and job completion rate constraints. Two system performance metrics are analyzed on top of the state transition diagram using Markov-chain theory. A numerical method is used to compute the non-linear model, and a genetic algorithm is used to solve the optimization model based on the double-element encoding mechanism. Illustrative examples are presented to explain the analysis of system transient availability and job completion rate as well as the allocation result under constraints. Experiment results indicate that with the same redundancy, the job completion rate of systems with the new mixed strategy is higher than the systems with traditional strategy. Thus, different redundancy should be allocated for different kinds of redundancy strategies, even under the same constraints.

Key words: redundancy allocation; reliability optimization; mixed redundancy strategy; genetic algorithm; series-parallel system

* 基金项目: 国家自然科学基金(61309005); 重庆市前沿与应用基础研究计划(cstc2014jcyjA40015)

Foundation item: National Natural Science Foundation of China (61309005); Frontier and Application Basic Research Program of Chongqing (cstc2014jcyjA40015)

收稿时间: 2015-09-18; 修改时间: 2015-12-02; 采用时间: 2016-01-26

时采用监控机制触发冷备份组件替换,发生故障的积极冗余组件一旦被探测到故障就被实时替换.虽然大型软件系统的组织架构较为复杂,但通常可采用基于体系结构的分析方法,通过系统组件的可靠性进行整体系统可靠性分析^[21].为简化体系架构运算的复杂性,对组件内部的冗余策略分析进行详细说明,本文采用串并联系统架构对基于组件的软件系统可靠性进行分析.在子系统内部对冗余组件采用马尔可夫链理论建立了新型混合冗余策略的可靠性及性能分析模型.由于该模型较为复杂,无法获得闭合形式的公式,数值化计算方法被用于求解该马尔可夫链中每个状态的实时概率.在对系统实时可用性以及任务执行效率进行分析的基础上建立了冗余分配模型,对冗余配置资源进行最小化,采用基于二元组编码的遗传算法对该模型进行求解.根据实例软件系统对基于混合冗余策略以及实时替换机制的系统分析过程和优化模型的求解过程进行了说明.

本文首先对串并联软件系统中支持组件实时替换混合冗余策略进行描述;其次针对该策略建立系统分析模型以及相应的优化模型;然后给出基于遗传算法的模型求解方法,并进行实例验证与分析;最后对本文内容进行总结,并对下一步工作进行展望.

1 支持实时替换的混合冗余策略

本文主要研究基于积极冗余机制以及冷备份冗余机制的混合冗余策略.假设在一个串并联系统中共有 m 个串联子系统,对每个子系统 $i(1 \leq i \leq m)$,共有 n_i 个并行运行或作为备份的组件. n_{Ai} 和 n_{Si} 分别表示每个子系统 i 中的积极冗余以及冷备份冗余组件数量: $n_i = n_{Ai} + n_{Si}$.上述基于混合冗余策略的串并联系统结构如图 1 所示.

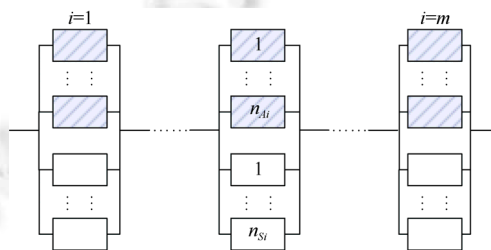


Fig.1 Structure of series-parallel system with mixed redundancy strategy

图 1 采用混合冗余策略的串并联系统结构

当采用传统混合冗余策略时,每个子系统假设初始共有 n_{Ai} 个积极冗余组件在正常运行,只有当所有的积极冗余组件失效时,才激活冷备份冗余组件.对每个子系统,一次仅采用 1 个冷备份冗余组件.但是在计算系统尤其是云计算系统中,失效的组件在被探测到失效后通常将被立即替换.例如,根据现有云计算系统中设计的容错策略^[6,17],一组主节点被用于任务执行,同时,一组次节点作为系统的冷备份存在.在任务执行过程中,监测进程将周期性地探测节点是否有故障发生.一旦主节点产生故障,迁移进程则将会被触发,该节点中的任务将会迁移至可用的次节点中继续执行.所以,冷备份冗余节点将不会等待所有的主节点失效后才被激活.当系统中执行任务的节点数量不满足任务需求时,执行的任务将会失败.根据可靠性分析的原理,对于具有串并联结构的软件系统,任意串联子系统的失效都将导致整体系统的失效,假设子系统中的多个组件始终运行同构任务并可随时进行任务迁移,则只有子系统中所有运行中组件均失效时子系统才会失效,所以每个子系统中满足任务执行需求的最少组件数量为 1.在子系统内部可采用多种冗余组件切换策略,在将上述容错策略应用于串并联系统中时,每个子系统初始仍有 n_{Ai} 个积极冗余组件在正常运行,同时,监控进程将周期性地探测组件状态并根据系统资源对失效组件进行替换.当系统中的所有冗余组件均失效时,子系统将失效并将引起系统的失效.由于监测进程执行的周期性,在某些时刻可能无法及时探测子系统内的失效组件,所以当子系统中正在运行的组件数量为 0 时,任务执行暂时失败,此时,子系统同样呈现不可用状态,但子系统中可能仍存在未能及时替换的冷备份冗余组件.对于任务可在组件间迁移的软件系统,如云计算软件,在重新替换冷备份冗余组件后,子系统仍可正常执行.在本文中主要考虑此类软件的运行情况,即当子系统内的冗余组件全部失效或正在运行中的组件全部失效且监

控进程未能及时探测时,判定该子系统为失效.上述机制的工作流程如图 2 所示.

与传统混合策略相比,新混合冗余策略将会产生不同的系统可靠性及系统性能,所以在每个子系统中需要分配的冗余度也不尽相同.下一节将对串并联系统中基于该策略的冗余分配问题做进一步的讨论.

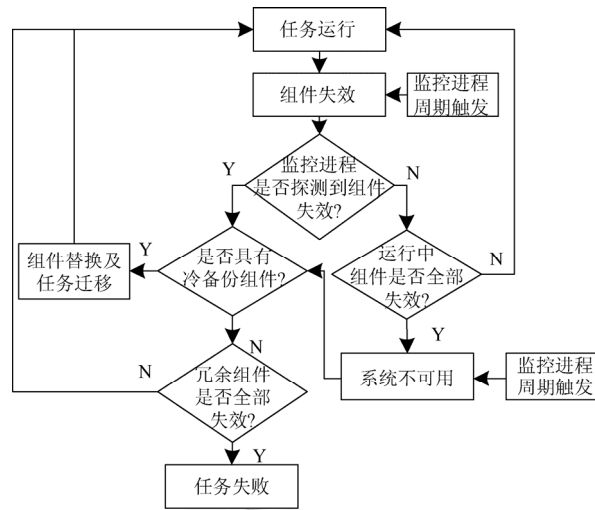


Fig.2 Mixed redundancy strategy with monitoring and instant switching

图 2 基于监测与实时替换的混合冗余策略

2 冗余分配优化模型

上一节中描述的支持组件实时替换的混合冗余机制与传统混合冗余策略有所不同,现有研究中还未对采用了新混合冗余策略的系统可靠性及性能进行分析.所以在提出该策略的冗余分配模型前,首先建立支持实时替换的混合冗余策略系统模型,并对性能指标进行分析.为了减小模型计算的复杂性,本文中选用串并联系统结构进行计算,对于其他复杂结构的软件系统,还可以采用基于体系结构的软件可靠性分析方法进行计算^[21].

2.1 系统模型

由于整体系统由多个串联子系统组成,系统可靠性及性能也受到子系统可靠性及性能的影响.所以,首先对每个子系统提出分析模型,以分析基于监测及实时替换的混合冗余策略.假设一个子系统中每个冗余组件的失效规律满足相同的分布,同时在任务迁移的过程中没有错误发生.基于无修复系统的假设,在时间 t ,基于传统混合冗余策略的子系统 i 的可靠性为^[16]

$$R_i(t) = (1 - (1 - r_i(t))^{n_{Ai}}) + \int_0^t r(t-u) f^{n_{Ai}}(u) du + \sum_{j=1}^{n_{Si}-1} \int_0^t \int_{t_1}^t r(t-u) f^j(u-t_1) f^{n_{Ai}}(t_1) du dt_1 \quad (1)$$

其中, $r_i(t)$ 为时间 t 子系统 i 中组件的可靠性,采用与时间相关的函数值进行描述; $f^j(t)$ 为第 j 次故障到达时间的概率密度函数,例如,该子系统 j 个独立分布的故障时间之和.虽然该模型中没有假定组件无故障工作时间满足特定分布,但采用类似概率分析方法对新型混合冗余策略可靠性进行分析,模型较为复杂,所以本文基于马尔可夫链理论建立了系统状态迁移模型用于系统分析.首先,针对上一节中对基于混合冗余策略的软件系统运行状态的描述(如图 2 所示)建立子系统内部状态迁移图.在构建状态迁移图前,参照文献[16]对系统做如下假设.

- (1) 在冷备份冗余组件被激活应用前,其失效状态及过程将不列入模型考虑.
- (2) 每次至多可对 y 个失效组件进行任务迁移,在模型中假设迁移全部成功,不考虑迁移失败的概率.
- (3) 在任务执行过程中不考虑组件的修复措施,即在模型中假设每个子系统中不具备修复设施.

(4) 每个子系统中每个组件的无故障工作时间满足相同的分布.

上述混合冗余策略的状态迁移图如图 3 所示.由于该图中具有较多的状态,为了避免状态爆炸,同时简化模型,对系统模型做进一步的两点假设.

(1) 子系统 i 中每个组件的无故障工作时间满足参数为 λ_i 的指数分布.

(2) 组件切换以及任务迁移的时间远小于两次监控之间的间隔时间.假设监控周期满足参数为 λ_{mi} 的指数分布,多个失效组件的替换过程并行执行,其中每个组件的替换时间满足参数为 μ_i 的指数分布.

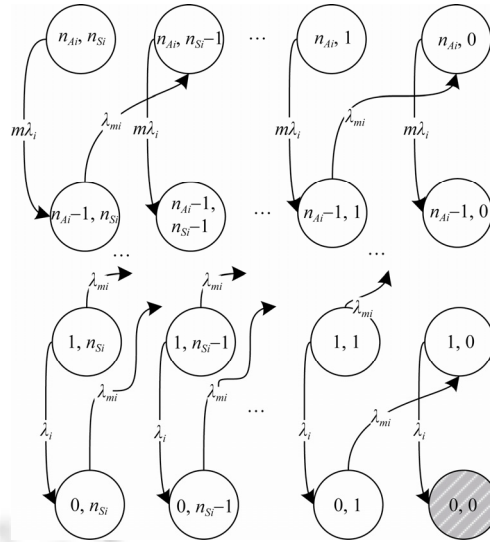


Fig.3 State transition diagram for mixed redundancy strategy with monitoring and instant switching

图 3 基于监测及实时替换的混合冗余策略状态迁移图

在图 3 中,状态 (j_{Ai}, j_{Si}) 表示当前子系统 i 中存在 $j_{Ai} (0 \leq j_{Ai} \leq n_{Ai})$ 个正常工作的组件以及 $j_{Si} (0 \leq j_{Si} \leq n_{Si})$ 个冷备份冗余组件.根据上一节对子系统运行流程的描述,子系统中对并联的组件最低需求量为 1,故仅状态 $(0, j_{Si})$ 表示子系统暂时或永久处于不可用状态.当 $j_{Si} > 0$ 时,一旦监测进程执行成功,子系统仍可从不可用状态恢复至正常状态,当 $j_{Si} = 0$ 时,则子系统无法恢复完全失效.所有的监测状态在该图中已被忽略.状态 $(j_{Ai} + 1, j_{Si})$ 到状态 (j_{Ai}, j_{Si}) 的迁移过程表示该子系统中任一积极工作组件发生故障.状态 (j_{Ai}, j_{Si}) 到状态 $(j_{Ai} + y, j_{Si} - y)$ 的迁移过程由两个迁移流程组合而成:周期性的监控探测流程以及 y 个失效组件的替换及任务迁移流程.从上述假设中可得知,该组合流程的时间满足参数为 λ_{mi} 和 μ_i 的二阶亚指数分布^[22].如果两次监测之间相隔的时间远大于组件替换的时间,则组合迁移过程可近似于满足参数为 λ_{mi} 的指数分布;否则,该过程可近似于满足参数为 μ_i 的指数分布.经过模型简化,图 3 中所有的状态迁移过程均满足指数分布,此时,系统整体流程可被视为一个连续时间马尔可夫链(CMTC).任意两个状态之间的迁移率是与时间无关的常数值.

在系统初始工作组件数目以及冷备份冗余组件数目为 n_{Ai} 和 n_{Si} 的前提下,假设 (j_{Ai}, j_{Si}) 和 (k_{Ai}, k_{Si}) 分别表示系统中的任意两个状态 $(0 \leq j_{Ai}, k_{Ai} \leq n_{Ai}, 0 \leq j_{Si}, k_{Si} \leq n_{Si})$, $Q_{(j_{Ai}, j_{Si})(k_{Ai}, k_{Si})}(n_{Ai}, n_{Si})$ 表示从状态 (j_{Ai}, j_{Si}) 到状态 (k_{Ai}, k_{Si}) 的迁移率.将 $Q_{(j_{Ai}, j_{Si})(k_{Ai}, k_{Si})}(n_{Ai}, n_{Si})$ 简写为 $Q_i(n_{Ai}, n_{Si})$, 其值为

$$Q_i(n_{Ai}, n_{Si}) = \begin{cases} j_{Ai} \lambda_i, & j_{Ai} = k_{Ai} + 1, j_{Si} = k_{Si} \\ \lambda_{mi}, & k_{Ai} - j_{Ai} = j_{Si} - k_{Si} = y, y = \min\{n_{Ai} - j_{Ai}, k_{Ai}\} \\ -i_1 \lambda_i - \lambda_{mi}, & j_{Ai} = k_{Ai}, j_{Si} = k_{Si}, 0 < j_{Si} \leq n_{Si} \\ -i_1 \lambda_i, & i_1 = i_2, j_1 = j_2 = 0 \\ -i_1 \lambda_i, & i_1 = i_2 = m, j_1 = j_2 = n \end{cases} \quad (2)$$

假设 $\mathbf{Q}_i(n_{Ai}, n_{Si})$ 表示 $Q_i(n_{Ai}, n_{Si})$ 的对应生成矩阵.由于矩阵 $\mathbf{Q}_i(n_{Ai}, n_{Si})$ 的结构较为复杂,以 $n_{Ai} = n_{Si} = 2$ 为例,假设

每次至多可对 2 个失效组件进行任务迁移,矩阵 $Q_i(2,2)$ 示例如下.

$$Q_i(2,2) = \begin{pmatrix} (2,2) & -2\lambda_i & 0 & 0 & 2\lambda_i & 0 & 0 & 0 & 0 & 0 \\ (2,1) & 0 & -2\lambda_i & 0 & 0 & 2\lambda_i & 0 & 0 & 0 & 0 \\ (2,0) & 0 & 0 & -2\lambda_i & 0 & 0 & 2\lambda_i & 0 & 0 & 0 \\ (1,2) & 0 & \lambda_{mi} & 0 & -\lambda_i - \lambda_{mi} & 0 & 0 & \lambda_i & 0 & 0 \\ (1,1) & 0 & 0 & \lambda_{mi} & 0 & -\lambda_i - \lambda_{mi} & 0 & 0 & \lambda_i & 0 \\ (1,0) & 0 & 0 & 0 & 0 & 0 & -\lambda_i & 0 & 0 & \lambda_i \\ (0,2) & 0 & 0 & \lambda_{mi} & 0 & 0 & 0 & -\lambda_{mi} & 0 & 0 \\ (0,1) & 0 & 0 & 0 & 0 & 0 & \lambda_{mi} & 0 & -\lambda_{mi} & 0 \\ (0,0) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3)$$

假设 $\pi_{(j_{Ai}, j_{Si})}(n_{Ai}, n_{Si}, t)$ 表示具有初始 n_{Ai} 和 n_{Si} 值的子系统在时间 t 正处于状态 (j_{Ai}, j_{Si}) 的实时概率,根据 Kolmogorov 的前向方程^[22],能够得到一组偏微分方程组:

$$\begin{cases} \frac{d\pi_{(n_{Ai}, n_{Si})}(n_{Ai}, n_{Si}, t)}{dt} = 0 - \pi_{(n_{Ai}, n_{Si})}(n_{Ai}, n_{Si}, t)n_{Ai}\lambda \\ \frac{d\pi_{(j_{Ai}, j_{Si})}(n_{Ai}, n_{Si}, t)}{dt} = \sum_{(k_{Ai}, k_{Si}) \in S} \pi_{(k_{Ai}, k_{Si})}(n_{Ai}, n_{Si}, t)Q_{(k_{Ai}, k_{Si})(j_{Ai}, j_{Si})}(n_{Ai}, n_{Si}, t) \\ \dots \\ \frac{d\pi_{(0,0)}(n_{Ai}, n_{Si}, t)}{dt} = \pi_{(1,0)}(n_{Ai}, n_{Si}, t)\lambda \end{cases} \quad (4)$$

假设 $\pi(n_{Ai}, n_{Si}, t)$ 表示多个状态 $\pi_{(j_{Ai}, j_{Si})}(n_{Ai}, n_{Si}, t)$ 的向量: $[\pi_{(n_{Ai}, n_{Si})}(n_{Ai}, n_{Si}, t), \dots, \pi_{(0,0)}(n_{Ai}, n_{Si}, t)]$, 则上述方程组可用如下矩阵形式表示:

$$\frac{d\pi(n_{Ai}, n_{Si}, t)}{dt} = \pi(n_{Ai}, n_{Si}, t)Q_i(n_{Ai}, n_{Si}, t) \quad (5)$$

求解该方程组即可得到 $\pi_{(j_{Ai}, j_{Si})}(n_{Ai}, n_{Si}, t)$ 的值. 根据上述计算流程,新型混合冗余策略的系统模型可由 $\pi_i(n_{Ai}, n_{Si}, t)$ 及 $Q_i(n_{Ai}, n_{Si}, t)$ 的值来描述.

2.2 可靠性及性能分析

虽然式(5)所示的方程组可以通过拉格朗日转换方法进行求解,但计算拉格朗日逆变换的难度较大.此外,在未给定 n_{Ai} 及 n_{Si} 的前提下,难以获得上述等式的数学闭合形式,所以也无法通过符号计算方法对上述问题进行直接求解.本节采用数值计算方法对实时概率值进行求解并获得了相应的可靠性及性能分析指标.在给定 n_{Ai} 及 n_{Si} 值的前提下,进一步将 $\pi_i(n_{Ai}, n_{Si}, t)$ 简写为 $\pi_i(t)$,将 $Q_i(n_{Ai}, n_{Si}, t)$ 简写为 Q_i . $\pi_i(t)$ 的值可通过随机方法计算^[23].

$$\pi_i(t) = \pi_i(0)e^{Q_i t}, e^{Q_i t} = \sum_{k=0}^{\infty} \frac{(Q_i t)^k}{k!} = \sum_{k=0}^{\infty} e^{-p_i t} \frac{(p_i t)^k}{k!} (Q_i^*)^k \approx \pi(0) \sum_{k=l_i}^{r_i} e^{-p_i t} \frac{(p_i t)^k}{k!} (Q_i^*)^k \quad (6)$$

其中, p_i 是矩阵 Q_i 对角向量元素绝对值中最大的值, $Q_i^* = Q_i/p_i + I$ 以消除矩阵 Q_i 中的负值并提高计算效率.给定可容忍误差值 ε , 式(6)中的无限序列求和可被截断为有限序列求和.根据误差容忍空间,可计算 l_i 及 r_i 的值为

$$\sum_{k=0}^{l_i-1} e^{-p_i t} \frac{(p_i t)^k}{k!} \leq \frac{\varepsilon}{2}, 1 - \sum_{k=0}^{r_i} e^{-p_i t} \frac{(p_i t)^k}{k!} \leq \frac{\varepsilon}{2} \quad (7)$$

由于子系统在运行之初具有 n_{Ai} 个积极冗余组件和 n_{Si} 个冷备份冗余组件,在时间 0, 子系统处于状态 (n_{Ai}, n_{Si}) 的概率为 1, 子系统处于其他任一状态的概率为 0. 所以, $\pi_i(0)$ 的初始值为 $[1, 0, \dots, 0]$. 在给定 n_{Ai} 及 n_{Si} 值后, $\pi_i(t)$ 的值可通过计算式(6)及式(7)得到.

根据广义可靠性的定义,软件系统可靠性可以通过多种指标进行度量,如多次调用中系统正常运行(没有失败)的概率^[21],在 t 时刻系统的失效次数及失效率^[24]等.考虑到本文主要分析长时间持续运行的软件系统,同时系统中不同子系统组件间可能具有异构特性,故首先选用随时间变化的函数值对系统可靠性进行描述.其次,考虑到在冗余组件替换过程中,系统失效与恢复状态的更迭,选用实时可用性对系统可靠性进行描述.实时可用性表

示任意时间 t , 子系统正常工作的概率. 每个子系统 i 的可靠性指标采用实时可用性 $A_i(n_{Ai}, n_{Si}, t)$ 进行评估. 由于仅有状态 $(0, j_{Si})$ 为系统不可用状态, 所以实时可用性为子系统处于可用状态时的概率值之和.

$$A_i(n_{Ai}, n_{Si}, t) = 1 - \sum_{j_{Si}=0}^{n_{Si}} \pi_{(0, j_{Si})}(n_{Ai}, n_{Si}, t) \quad (8)$$

当子系统中正在运行的冗余组件数量不同时, 子系统的任务处理速度不尽相同. 假设子系统每个组件除了具有相同的失效规律外, 其工作效率和配置代价也相同. 由于子系统中存在多种工作状态, 为了表示一段时间内的软件系统性能, 采用子系统的平均任务执行率进行描述. 平均任务执行率为子系统在各个工作状态下体现的任务执行率与其处于该工作状态的概率值平均和. 假设子系统 i 中每个组件的任务执行效率为 r_i , 则具有 n_{Ai} 个积极冗余组件的子系统任务执行效率为 $n_{Ai}r_i$. 假设 $G_i(n_{Ai}, n_{Si}, t)$ 表示时间 t 子系统 i 的实时任务执行率, $P_i(n_{Ai}, n_{Si}, t)$ 表示相应的平均任务执行率, 其值为 $G_i(n_{Ai}, n_{Si}, t)$ 在时间 $0 \sim$ 时间 t 间隔时间内的均值. $P_i(n_{Ai}, n_{Si}, t)$ 的值可通过各个状态任务执行率的平均值计算, 其表示子系统的性能, $P_i(n_{Ai}, n_{Si}, t)$ 值为

$$P_i(n_{Ai}, n_{Si}, t) = E(G_i(n_{Ai}, n_{Si}, t)) = \sum_{(j_{Ai}, j_{Si}) \in S} j_{Ai} r_i \pi_{(i, j)}(n_{Ai}, n_{Si}, t) \quad (9)$$

假设子系统中每个组件配置资源的代价为 c_i , 则子系统的总体代价 $C_i(n_{Ai}, n_{Si})$ 为其配置的所有冗余资源代价之和.

$$C_i(n_{Ai}, n_{Si}) = c_i(n_{Ai} + n_{Si}) \quad (10)$$

2.3 冗余分配模型

根据式(8)~式(10)的评估, 每个子系统的可靠性及性能都可通过其配置的初始积极冗余度值及冷备份冗余度值表示. 所以, 为每个子系统选择合适的冗余度值将有益于提高系统的可靠性及性能, 同时能够降低冗余组件配置所消耗的资源. 本文所提出的冗余分配模型旨在当满足系统可靠性及性能的约束条件时达到冗余资源配置代价的最小化.

$$\begin{cases} \min C_{\text{sys}} = \sum_{i=1}^m C_i(n_{Ai}, n_{Si}) \\ \text{s.t. } A_{\text{sys}} = \prod_{i=1}^m A_i(n_{Ai}, n_{Si}, t) \geq A_0, \\ P_{\text{sys}} = 1 / \sum_{i=1}^m (1/P_i(n_{Ai}, n_{Si}, t)) \geq P_0, 1 \leq n_{Ai} \leq n_0, 0 \leq n_{Si} \leq n_0 \end{cases} \quad (11)$$

其中, A_{sys} 表示系统的实时可用性, C_{sys} 表示系统整体冗余配置代价, P_{sys} 表示系统整体任务执行效率. 由于串联系统的可靠性为其内部子系统可靠性的乘积, 所以 A_{sys} 可以通过计算各个子系统实时可用性 $A_i(n_{Ai}, n_{Si}, t)$ 的乘积得到. C_{sys} 为系统冗余配置的所有代价之和, 也即每个子系统冗余配置代价 $C_i(n_{Ai}, n_{Si})$ 之和. 假设单个子系统任务执行效率与任务执行时间的乘积为 1, 则 $1/P_i(n_{Ai}, n_{Si}, t)$ 表示每个子系统的平均任务执行时间. 由于串联系统的任务执行时间为所有子系统任务执行时间的总和, 所以 P_{sys} 采用系统任务执行时间总和的倒数表示. A_0 与 P_0 分别是系统可靠性及性能约束指标, n_0 是子系统中每种冗余策略可配置的最大冗余组件数目. 每个子系统中初始积极冗余以及冷备份冗余两种冗余策略的值是该分配模型中的变量.

在初始冗余度值未确定的情况下, 上一节中已说明子系统可靠性及性能不具有闭合形式的符号表达式, 所以该优化模型也不具备闭合形式的符号表达式.

3 优化模型求解算法

虽然冗余度优化模型中的所有变量均为整数值, 但该模型并不是传统具有闭合形式的非线性优化模型, 也难以采用整型规划的方法进行求解. 所以在本节中采用遗传算法搜索近似最优解.

3.1 遗传算法框架

遗传算法是解决组合优化问题的启发式搜索算法之一,并已被应用于与 RAP 相关的大量研究中^[25,26],该算法的基本框架如下.

Step I. 设置子种群 $Q_0 = \emptyset$, 种群迭代计数器 $t=0$. 采用 N 个随机生成个体对父种群进行初始化: $P_0 = \{p_1, p_2, \dots, p_N\}$.

Step II. 当 $t < t_{\max}$ 时, 完成以下步骤.

1. 采用遗传算子从父种群 P_t 产生子种群 Q_t .
2. 合并上一步骤中的父子种群, 产生新种群 $R_t = P_t \cup Q_t$.
3. 根据目标函数值以及约束函数值计算种群 R_t 中每个个体的适应度值.
4. 按照适应度值将 R_t 中的个体倒序排列.
5. 采用轮盘赌选择方法从 R_t 中选择 N 个个体并放入下一代父种群 P_{t+1} 中.
6. 设置 $t=t+1$ 并返回 Step II.

Step III. 当种群迭代次数超过最大值时停止算法.

按照类似的算法结构, 遗传算法中需要设计的部分包括解编码机制、遗传算子和适应度计算方法. 以上 3 部分的具体实现流程分别参见以下 3 小节.

3.2 解编码机制

在遗传算法的每次迭代中, 种群中的每个个体都代表了冗余分配问题的一个特定解. 该模型中有 $2 \times m$ 个未知向量需要求解, 参照现有的三元组编码方法^[5,16], 本文中采用了双元组编码机制对冗余分配问题的解进行编码. 算法中的每个解都由一个 $2 \times m$ 的整型数组表示. 图 4 中展示了当 $m=6$ 时, 采用该编码机制的一个解的例子.

| | | | | | | | |
|-------|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| 积极冗余 | 1 | 2 | 1 | 3 | 4 | 5 | 2 |
| 冷备份冗余 | 2 | 1 | 1 | 4 | 2 | 3 | 1 |

Fig.4 Example of double-element encoding mechanism
图 4 双元组编码机制示例

图 4 中的矩阵第 1 行表示每个子系统中积极冗余机制的冗余度, 第 2 行表示每个子系统冷备份冗余机制的冗余度. 矩阵中的每一列表示特定子系统混合冗余机制的整体配置方案. 例如, 根据图 4 所示的配置, 第 1 个子系统具有 2 个积极冗余备份组件而最后一个子系统包含 1 个冷备份冗余组件.

3.3 遗传算子

基于双元组编码方式设计了相应的遗传算子. 在交叉操作中, 首先生成一个 $2 \times m$ 大小的交叉掩阵, 在该矩阵中只包含 0 和 1 两种元素, 同时该矩阵由两类元素随机组成, 如图 5 所示. 对于任意选择的两个个体 p_i 和 p_j , 如果交叉掩阵中的某元素为 1, 则两个解矩阵中对应位置的元素将进行互换操作, 交叉操作的示例也如图 5 所示.

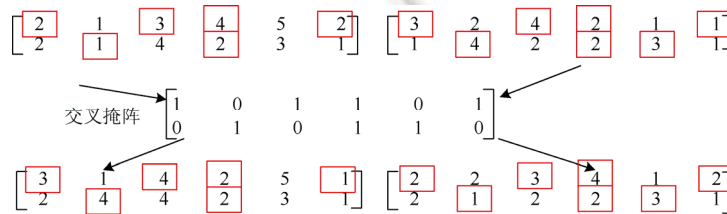


Fig.5 Example of crossover operation for double-element encoded individuals
图 5 双元组编码的个体交叉操作示例

在变异操作中,类似 f 也通过 0 和 1 元素随机生成一个 $2 \times m$ 大小的变异掩阵.变异掩阵与交叉掩阵不同的是,在该矩阵的每一行中,至多两个元素能够为 1,其他元素均为 0,如图 6 所示.该规则的设置主要是为了避免在变异过程中对个体造成剧烈的变化.对任一随机选择的个体 p_i ,如果变异掩阵中的某元素为 1,则将对 p_i 中相应位置的元素进行变换.变换过程与种群初始化过程中每个个体每个元素的生成过程类似.对积极冗余度,将从 $1 \sim n_0$ 中随机选择一个整数对该元素值进行替换;对冷备份冗余度,将从 $0 \sim n_0$ 中随机选择一个整数进行替换.

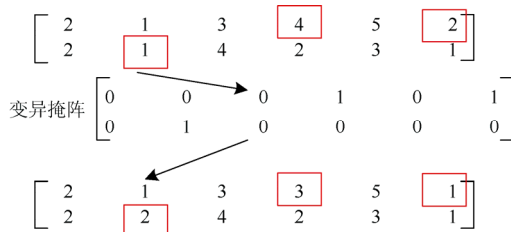


Fig.6 Example of mutation operation for double-element encoded individuals
图 6 双元组编码的个体变异操作示例

3.4 适应度值计算方法

由于式(11)所示的 RAP 问题是单目标多约束优化问题,每个个体的适应度值 I 采用满足约束条件下的目标值大小来表示.

$$I = \begin{cases} C_{\text{sys}}, & P_{\text{sys}} \geq P_0 \ \& \ A_{\text{sys}} \geq A_0 \\ \infty, & \text{else} \end{cases} \quad (12)$$

4 实例验证

本节采用实例系统说明基于监测与实时替换机制的混合冗余策略优化分配模型及其求解过程,同时,将优化后支持实时替换的混合冗余策略与传统的混合冗余策略对系统产生的不同影响进行对比分析.

4.1 系统可靠性及性能分析

本实验中采用现有研究已使用的实例软件系统^[16]:一个具有 8 个子系统的串并联系统.在每个子系统中没有考虑存在多种选择的组件而只考虑存在一种相同类型的组件.每个子系统中组件的失效率以及配置资源代价见表 1.每个组件的监控频率 λ_{mi} 均被设置为 1 次/小时.每个子系统中组件的任务执行效率为 1 任务单元.式(7)序列缩减的过程中所容许的误差值 ϵ 为 0.001.首先,对系统可靠性及性能的分析模型进行了实验.

Table 1 Component configuration parameters for example system

表 1 示例系统的组件配置参数

| 组件编号 | λ_i/h^{-1} | c_i | 组件编号 | λ_i/h^{-1} | c_i |
|------|--------------------|-------|------|--------------------|-------|
| 1 | 0.004 99 | 2 | 5 | 0.004 31 | 2 |
| 2 | 0.008 18 | 2 | 6 | 0.005 67 | 3 |
| 3 | 0.004 66 | 4 | 7 | 0.010 5 | 3 |
| 4 | 0.006 83 | 5 | 8 | 0.015 0 | 3 |

表 2 中给出了一组随机生成的子系统混合冗余度值.随着时间 t 的增加,整个串并联系统中的可靠性及性能指标如图 7 所示.为了比较本文中采用的新型混合冗余策略和传统的混合冗余策略,根据同一组参数,对采用了传统混合冗余策略的系统指标也进行了评估,其可靠性及性能结果也如图 7 所示.图 7 中的横坐标表示时间 t 的增长,图 7(a)和图(b)中的纵坐标分别表示系统可靠性 A_{sys} (图中简写为 A)以及性能 P_{sys} (图中简写为 P).图 7 中的曲线分别代表传统混合冗余策略(traditional)以及支持实时替换的混合冗余策略(instant switching).

图 7 中的结果显示,当采用同样的冗余资源时,采用传统混合冗余策略的系统可靠性优于支持实时替换的

混合冗余策略系统,而后者的性能比前者更优.上述结果表明,采用同样的冗余资源,不同的混合策略配置和方法将使系统产生不同的可靠性及性能指标.进一步对结论进行了实验,将时间 t 设置为 50 小时,改变系统中第 4 个子系统的积极冗余度和冷备份冗余度,从 1 变化至 10.采用了不同混合冗余策略的系统可靠性及性能计算结果如图 8 所示.与图 7 类似的是,图 8 中的纵坐标分别表示系统可靠性 A_{sys} 以及性能 P_{sys} ,但图 8 中的横坐标依次表示第 4 个子系统的初始积极冗余度 n_{Ai} 及冷备份冗余度 n_{Si} .

Table 2 Mixed strategy redundancy example

表 2 混合冗余策略冗余度示例

| 组件编号 | n_{Ai} | n_{Si} | 组件编号 | n_{Ai} | n_{Si} |
|------|----------|----------|------|----------|----------|
| 1 | 2 | 4 | 5 | 2 | 4 |
| 2 | 1 | 2 | 6 | 2 | 3 |
| 3 | 1 | 1 | 7 | 3 | 1 |
| 4 | 3 | 2 | 8 | 2 | 2 |

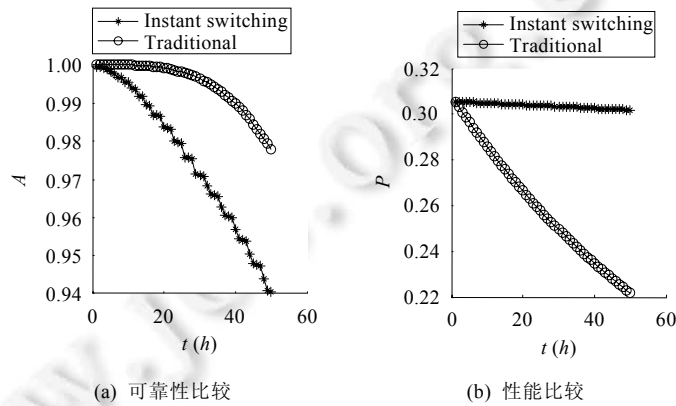


Fig.7 Reliability and performance of system using mixed redundancy strategies changing with time
图 7 基于不同混合冗余策略的系统可靠性及性能随时间变化规律

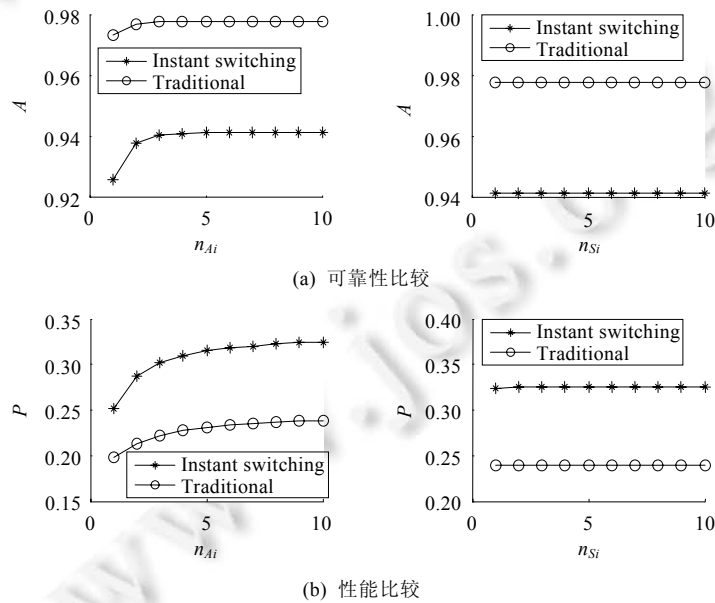


Fig.8 Reliability and performance of system using mixed redundancy strategies changing redundancy
图 8 基于不同混合冗余策略的系统可靠性及性能随冗余度变化规律

在图 8 中,积极冗余的变化将对两种策略所产生的可靠性及性能的变化产生不同的影响.当改变系统冷备份冗余度时,系统指标产生的变化较小.为对冗余度变化时系统指标的变化进行具体分析,每个实验案例中冗余度变化的影响如图 9 所示.图 9 中的纵坐标分别表示系统可靠性的变化 ΔA 以及性能的变化 ΔP .图 9 的结果表明:当采用支持实时替换的混合冗余策略时,冗余度的变化将对系统产生更大的影响.综上所述,即使给定相应的系统约束,对不同的混合冗余策略也需要进行不同的冗余度配置.此外,图 9 中的结果还显示:当采用传统混合冗余机制时,冷备份冗余度的变化对系统性能未产生任何影响,这也正是支持实时替换的混合冗余策略被引入的原因.

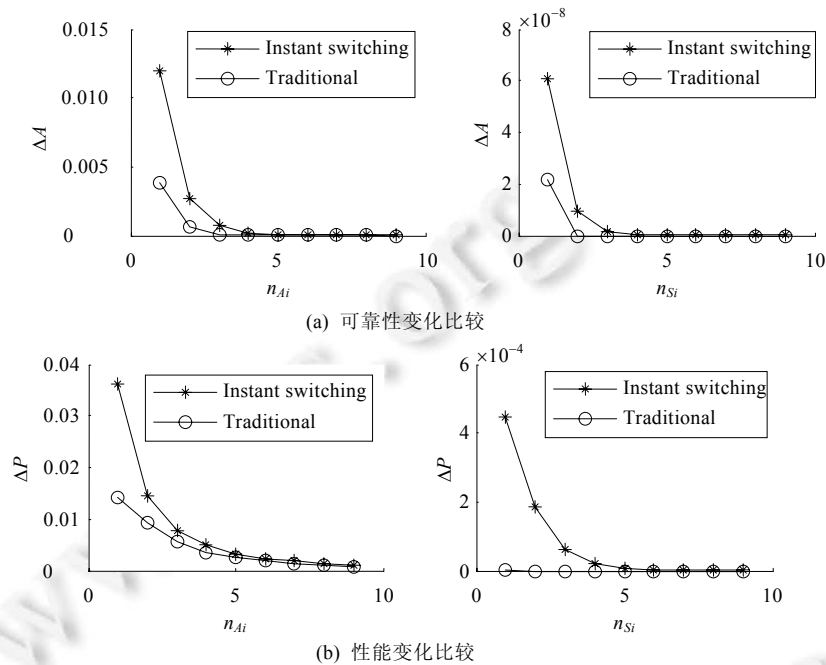


Fig.9 Impact of redundancy change on the reliability and performance change using mixed redundancy strategies
图 9 基于不同混合冗余策略的系统中冗余度变化对可靠性及性能变化影响分析

4.2 冗余分配比较

上一节的实验结果表明:随着冗余度的增加,冗余度的变化对系统可靠性及性能的影响将逐步降低.所以,当冗余度增大到一定程度后,其变化对系统影响较小,在本节中,任一子系统中积极冗余或冷备份冗余的最大值被设为 5.系统时间 t 被设为 50 小时.经过多次算法实验比较,算法中最大种群代数为 40,种群中个体数量为 50.假设系统实时可用性约束为 0.999 9,任务执行效率约束为 0.25.基于以上参数,采用支持实时替换的混合冗余策略,使用遗传算法求解最优冗余度值,算法每次迭代时种群中最好的适应度值如图 10 所示.图 10 中的横坐标表示种群代数,纵坐标表示每一代种群的适应度值.图 10 中的结果显示,算法在大约 20 代种群时即收敛.

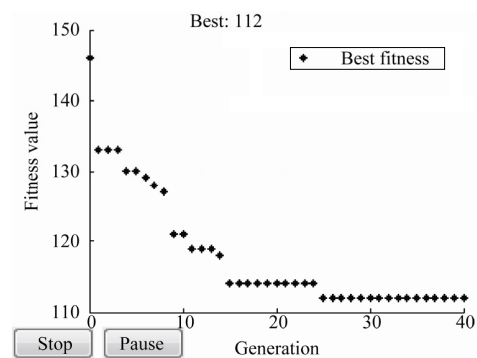


Fig.10 Best fitness value of each generation
图 10 每代种群的最好适应度值

在本实验中,所获得的最优冗余配置代价为 112,相应的每个子系统的冗余度见表 3.表 3 中也给出了采用传统混合冗余策略的每个子系统冗余度值,采用该策略,所有冗余度配置代价为 106.

Table 3 Allocation result for two types of mixed redundancy strategy

表 3 两种混合冗余策略的冗余分配结果

| 组件编号 | 支持实时替换的混合冗余策略 | | 传统混合冗余策略 | |
|------|---------------|----------|----------|----------|
| | n_{Ai} | n_{Si} | n_{Ai} | n_{Si} |
| 1 | 3 | 2 | 2 | 2 |
| 2 | 2 | 3 | 4 | 1 |
| 3 | 2 | 2 | 2 | 1 |
| 4 | 2 | 2 | 3 | 1 |
| 5 | 2 | 2 | 3 | 1 |
| 6 | 2 | 3 | 3 | 2 |
| 7 | 2 | 3 | 3 | 2 |
| 8 | 2 | 4 | 4 | 2 |

在下面的实验中,对系统时间 t 和系统约束进行了变化,分别计算两种策略相应的冗余分配结果.由于遗传算法是一种随机搜索算法,每次实验可能生成不同的近似最优解,所以对每次实验分别执行了 5 次算法,将 5 次算法中计算的最优解作为最终解.首先在性能约束=0.3,可靠性约束=0.99 的前提下,将系统时间 t 从 10 小时增加到 50 小时,相应的冗余分配结果如图 11(a)所示.该图的横坐标为时间变化的趋势,纵坐标为系统总体代价 C_{sys} (图中简称为 C)的变化.然后设定系统时间为 50 小时,可靠性约束为 0.99,将性能约束从 0.25 变化至 0.5,每种配置策略相应的冗余度配置代价如图 11(b)所示.图中的横坐标为性能约束值 P_0 的变化.之后,将性能约束分别设定为 0.25 及 0.3,系统时间为 50 小时,在可靠性约束从 0.95 到 0.99 变化的过程中,相应的冗余配置代价分别如图 11(c)和图 11(d)所示.上述两图的横坐标均为可靠性约束值 A_0 的变化,纵坐标与前图相同.

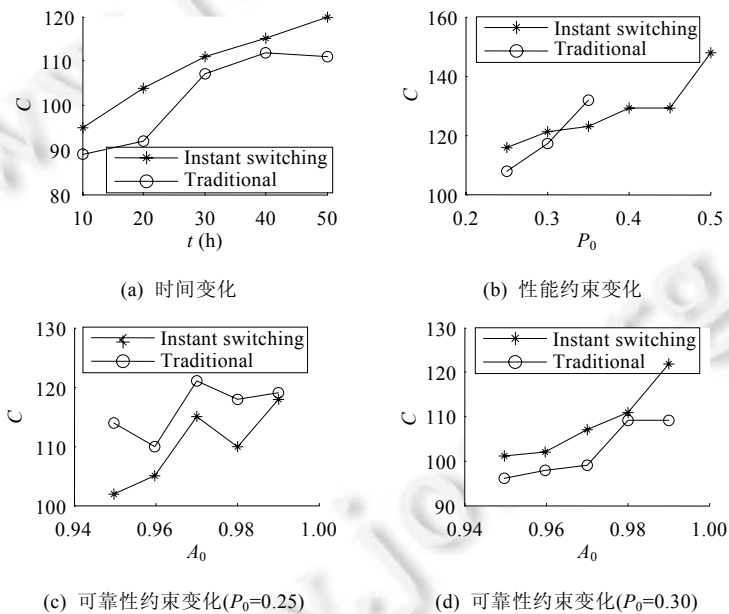


Fig. 11 Change of redundancy cost of two mixed redundancy strategies with the change of constraints and time t

图 11 随着时间 t 以及系统约束值的变化,两种混合冗余配置策略相应的冗余度配置代价变化

图 11 中的结果表明,当系统性能约束相对较低(<0.3)时,传统策略配置所消耗的冗余代价比支持实时替换的新策略消耗的代价少,但是,随着性能约束值的增加,传统策略所消耗的代价将超过本文中的新策略.当性能约束值超过 0.35 时,传统的冗余配置策略已经难以获得最优冗余度从而达到系统约束要求,而支持实时替换的

新策略能够获得最优的冗余度配置.

5 结论及展望

虽然现有研究已对系统中的多种冗余配置策略以及相应的混合配置策略进行了讨论,但它们基本假设只有当所有积极冗余组件失效时才启用冷备份冗余组件.但是,在需要长时间持续运行的软件系统,如云计算系统中,对系统持续运行时间和任务响应速度的需求增加使得对系统可靠性及性能保障同等重要.在此类系统中,采用基于监控替换的机制,一旦监测到积极冗余组件出错,备份组件将马上被启用,否则系统性能将受到一定的影响.针对性能与可靠性同等重要的系统,考虑采用支持组件实时替换机制的混合冗余策略,提出了针对新混合冗余配置策略的冗余度分配模型.基于马尔可夫链理论首先构建了子系统状态迁移图,并用于系统分析,采用了实时可用性评估系统可靠性,采用任务完成效率评估系统性能,通过数值计算方法对上述指标进行计算.在此基础上建立了冗余度分配模型以便在系统可靠性及性能约束的条件下对系统冗余度代价进行最小化.由于该分配模型不是传统的具有闭合形式的非线性优化模型,所以采用遗传算法对该模型进行求解,并使用实例系统对求解过程进行说明.实验结果表明:采用新型混合冗余配置策略的系统评估指标值与采用传统混合策略的系统有较大的不同,所以即使在相同的约束条件下,针对不同类型的系统也需要配置不同的冗余度.在相同的实验条件下,当系统性能约束较高时,无法通过冗余度的优化配置使传统的混合冗余配置策略达到系统约束要求,但是采用新型的混合策略,能够通过冗余度的优化满足系统需求.

本文对支持实时替换的混合冗余配置策略中的冗余度优化分配进行了初步研究,但该研究的模型基于较多假设条件构建,例如假设子系统中的每个组件无故障工作时间均满足指数分布.假设条件可能影响模型对实际运行系统真实状态的描述,所以,在未来的研究中将对系统运行环境进行更通用的假设,并进一步改进系统评估模型.同时,也需要在遗传算法中增加自定义的局部搜索算子以提高算法搜索效率.

References:

- [1] Qiu W, Zheng Z, Wang X, Yang X, Lyu MR. Reliability-Based design optimization for cloud migration. *IEEE Trans. on Services Computing*, 2014,7(2):223–236. [doi: 10.1109/tsc.2013.38]
- [2] Kuo W, Wan R. Recent advances in optimal reliability allocation. *IEEE Trans. on Systems, Man, and Cybernetics Part A: Systems and Humans*, 2007,37(2):143–156. [doi: 10.1109/tsmca.2006.889476]
- [3] Ardakan MA, Hamadani AZ. Reliability optimization of series-parallel systems with mixed redundancy strategy in subsystems. *Reliability Engineering & System Safety*, 2014,130:132–139. [doi: 10.1016/j.res.2014.06.001]
- [4] Chern MS. On the computational complexity of reliability redundancy allocation in a series system. *Operations Research Letters*, 1992,11(5):309–315. [doi: 10.1016/0167-6377(92)90008-q]
- [5] Tavakkoi-Moghaddam R, Safari J, Sassani F. Reliability optimization of series-parallel systems with a choice of redundancy strategies using a genetic algorithm. *Reliability Engineering & System Safety*, 2008,93(4):550–556. [doi: 10.1016/j.res.2007.02.009]
- [6] Yang CT, Liu JC, Hsu CH, Chou WL. On improvement of cloud virtual machine availability with virtualization fault tolerance mechanism. *Journal of Supercomputing*, 2014,69(3):1103–1122. [doi: 10.1007/s11227-013-1045-1]
- [7] Zhang Z, Xiao L, Zhu M, Ruan L. Mvmotion: A metadata based virtual machine migration in cloud. *Cluster Computing*, 2014,17(2):441–452. [doi: 10.1007/s10586-013-0245-z]
- [8] Caserta M, Voss S. An exact algorithm for the reliability redundancy allocation problem. *European Journal of Operational Research*, 2015,244(1):110–116. [doi: 10.1016/j.ejor.2015.01.008]
- [9] Liu Y, Huang HZ, Wang Z, Li Y, Yang Y. A joint redundancy and imperfect maintenance strategy optimization for multi-state systems. *IEEE Trans. on Reliability*, 2013,62(2):368–378. [doi: 10.1109/tr.2013.2259193]
- [10] Coit DW, Liu J. System reliability optimization with k -out-of- n subsystems. *Int'l Journal of Reliability, Quality and Safety Engineering*, 2000,7(2):129–142. [doi: 10.1142/S0218539300000110]

- [11] Coit DW. Cold-standby redundancy optimization for nonrepairable systems. *IIE Trans.*, 2001,33(6):471–478. [doi: 10.1023/a:1007689912305]
- [12] Coit DW. Maximization of system reliability with a choice of redundancy strategies. *IIE Trans.*, 2003,35(6):535–543. [doi: 10.1080/07408170304420]
- [13] Chambari A, Rahmati SHA, Najafi AA, Karimi A. A bi-objective model to optimize reliability and cost of system with a choice of redundancy strategies. *Computers and Industrial Engineering*, 2012,63(1):109–119. [doi: 10.1016/j.cie.2012.02.004]
- [14] Chambari A, Najafi AA, Rahmati SHA, Karimi A. An efficient simulated annealing algorithm for the redundancy allocation problem with a choice of redundancy strategies. *Reliability Engineering and System Safety*, 2013,119:158–164. [doi: 10.1016/j.res.2013.05.016]
- [15] Safari J. Multi-Objective reliability optimization of series-parallel systems with a choice of redundancy strategies. *Reliability Engineering and System Safety*, 2012,108:10–20. [doi: 10.1016/j.res.2012.06.001]
- [16] Ardakan MA, Hamadani AZ. Reliability optimization of series-parallel systems with mixed redundancy strategy in subsystems. *Reliability Engineering and System Safety*, 2014,130:132–139. [doi: 10.1016/j.res.2014.06.001]
- [17] Mohamed N, Al-Jaroodi J. MidCloud: An agent-based middleware for effective utilization of replicated Cloud services. *Software-Practice and Experience*, 2015,45(3):343–363. [doi: 10.1002/spe.2235]
- [18] Cores I, Rodriguez G, Martin MJ, Gonzalez P, Osorio RR. Improving scalability of application-level checkpoint-recovery by reducing checkpoint sizes. *New Generation Computing*, 2013,31(3):163–185. [doi: 10.1007/s00354-013-0302-4]
- [19] Cui X, Mills B, Znati T, Melhem R. Shadow replication: An energy-aware, fault-tolerant computational model for green cloud computing. *Energies*, 2014,7(8):5151–5176. [doi: 10.3390/en7085151]
- [20] Levitin G, Xing LD, Johnson BW, Dai YS. Mission reliability, cost and time for cold standby computing systems with periodic backup. *IEEE Trans. on Computers*, 2015,64(4):1043–1057. [doi: 10.1109/tc.2014.2315644]
- [21] Pietrantuono R, Russo S, Trivedi KS. Software reliability and testing time allocation: An architecture-based approach. *IEEE Trans. on Software Engineering*, 2010,36(3):323–337. [doi: 10.1109/tse.2010.6]
- [22] Trivedi KS. *Probability and Statistics with Reliability, Queuing, and Computer Science Applications*. 2nd ed., New York: John Wiley and Sons, 2001.
- [23] Reibman A, Trivedi K. Numerical transient analysis of Markov-models. *Computers & Operations Research*, 1988,15(1):19–36. [doi: 10.1016/0305-0548(88)90026-3]
- [24] Goel AL, Okumoto K. Time-Dependent error-detection rate model for software reliability and other performance-measures. *IEEE Trans. on Reliability*, 1979,28(3):206–211. [doi: 10.1109/TR.1979.5220566]
- [25] Levitin G, Lisnianski A, Ben-Haim H, Elmakis D. Redundancy optimization for series-parallel multi-state systems. *IEEE Trans. on Reliability*, 1998,47(2):165–172. [doi: 10.1109/24.722283]
- [26] Tian Z, Levitin G, Zuo MJ. A joint reliability-redundancy optimization approach for multi-state series-parallel systems. *Reliability Engineering and System Safety*, 2009,94(10):1568–1576. [doi: 10.1016/j.res.2009.02.021]



何盼(1984—),女,重庆人,博士,副研究员,CCF 专业会员,主要研究领域为可信计算,可靠性分析及优化.



袁月(1986—),女,助理研究员,主要研究领域为优化控制计算.



郑志浩(1988—),男,助理研究员,主要研究领域为无线通信与嵌入式系统分析.



谭春(1974—),男,工程师,主要研究领域为机械系统可靠性分析.