# A Multi-objective Approach to Redundancy Allocation Problem in Parallel-series Systems

Zai Wang, Tianshi Chen, Ke Tang, and Xin Yao

***Abstract*— The Redundancy Allocation Problem (RAP) is a kind of reliability optimization problems. It involves the selection of components with appropriate levels of redundancy or reliability to maximize the system reliability under some predefined constraints. We can formulate the RAP as a combinatorial problem when just considering the redundancy level, while as a continuous problem when considering the reliability level. The RAP employed in this paper is that kind of combinatorial optimization problems. During the past thirty years, there have already been a number of investigations on RAP. However, these investigations often treat RAP as a single objective problem with the only goal to maximize the system reliability (or minimize the designing cost). In this paper, we regard RAP as a multi-objective optimization problem: the reliability of the system and the corresponding designing cost are considered as two different objectives. Consequently, we can utilize a classical Multi-objective Evolutionary Algorithm (MOEA), named Non-dominated Sorting Genetic Algorithm II (NSGA-II), to cope with this multi-objective redundancy allocation problem (MORAP) under a number of constraints. The experimental results demonstrate that the multi-objective evolutionary approach can provide more promising solutions in comparison with two widely used single-objective approaches on two parallel-series systems which are frequently studied in the field of reliability optimization.**

## I. Introduction

The Redundancy Allocation Problem (RAP) [1]–[4] is one of the most important reliability optimization problems in the designing phase of the parallel-series systems, network systems, *k-out-of-n:G* systems and other systems with various structures. For a system, the reliability of it can be increased by properly allocating redundancies to its subsystems [5], so RAP is formulated involving the selection of components with the appropriate levels of redundancy to maximize the system reliability under some predefined constraints. RAP is difficult to cope with because of its enormous requirement of computational time to find an optimal solution (as a NP-hard problem). In the field of system-reliability [6], [7], it has been reported that various single-objective optimization techniques, such as dynamic programming, integer programming, meta-heuristic algorithm, mixed integer and nonlinear programming and genetic algorithm [8]–[15], have been used to cope with RAPs. Although these techniques have their own advantages on RAPs, they all treat RAPs as single-objective problems, and the only goal is to maximize system reliability (or minimize the designing cost of the system).

Some researchers have also noticed that determining the redundancy allocation of the system should take multiple considerations into account, e.g., one hopes to obtain a system with *high reliability* while he or she will certainly prefer spending *low cost* in the designing phase of the system. Concretely, they considered both the reliability and cost [16]–[21] by aggregating the two objectives (reliability and cost) to a unique scalar objective function, and optimize the new objective function via some single-objective optimization technique. The above investigations have taken important steps towards finding more effective and efficient approaches for RAP. However, for single-objective approaches, one has to design sophisticated mechanisms of combining different objectives so as to achieve promising performances. On the other hand, the aggregation of two objectives may eliminate the possibility of finding multiple non-dominated solutions, which would leave less choices for system designer in practice. To cope with the above two difficulties, using some multi-objective approaches (such as some well-known Multi-Objective Evolutionary Algorithms (MOEAs) might be an appropriate choice. There have been some similar attempts for the reliability optimization [22]–[24]. In [22], a so-called MOMS-GA was proposed to solve the tri-objective redundancy allocation problem in multi-state systems, where the availability, cost and weight of the systems are considered as the three objectives. In [23], a tabu search meta-heuristic approach is first utilized to solve a bi-objective (reliability and cost) redundancy allocation problem. And then, based on the obtained Pareto optimal solutions, Monte-Carlo simulation is employed provide a decision maker with a pruned and prioritized set of Pareto-optimal solutions based on user-defined objective function preferences. In [24], a problem-specific MOEA is employed to solve the continuous reliability optimization problems (single-objective or bi-objective) on anomalistic complex systems, wherein the reliability of the components are variables to be optimized. In this paper, we utilize an efficient second-generation MOEA (NSGA-II) to solve the combinational redundancy allocation problem on parallel-series systems.

In this paper, we also formulate the RAP by considering the system reliability and designing cost as two objectives, and the resultant Multi-Objective RAP (MORAP) takes both objectives into account simultaneously. Solving the above MORAP by MOEAs can provide the designers more solutions with respect to different levels of trade-off between the two objectives while traditional single-objective approaches

The authors are with the Nature Inspired Computation and Applications Laboratory, the Department of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China. Xin Yao is also with CERCIA, the school of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, U.K.(emails: wangzai@mail.ustc.edu.cn, cetacy@mail.ustc.edu.cn, ketang@ustc.edu.cn, x.yao@cs.bham.ac.uk).

978-1-4244-2959-2/09/$25.00 © 2009 IEEE

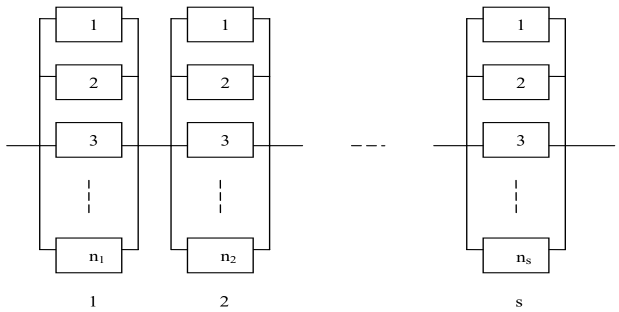


Fig. 1. The structure of parallel-series system

[5], [10] can only provide a unique solution at a time. Concretely, we utilize a well-known multi-objective optimization algorithm named Non-dominated Sorting Genetic Algorithm II (NSGA-II) to solve the MORAP in our empirical investigation. Efficacy of NSGA-II is demonstrated on two reliability growth parallel-series systems, one contains 5 subsystems while the other consists of 7 subsystems. The experimental results shows that NSGA-II offers a number of promising solutions, which enables the designers to consider the trade-off between the system reliability and designing cost.

The rest of this paper is organized as follows: Section 2 presents the single-objective redundancy allocation problem (SORAP) and multi-objective redundancy allocation problem (MORAP) on parallel-series system; Section 3 shows how we solve the redundancy allocation problem with our problem-specific NSGA-II. In Section 4, our multi-objective approach is experimentally studied in comparison with some widely used approaches on two parallel-series software systems. Section 5 contains our conclusion and future work.

## II. SORAP and MORAP on Parallel-series System

### A. Description of Parallel-series System

In this paper, we consider the RAP on the parallel-series systems which have already received intensive investigations [2], [5], [10]. A typical structure of parallel-series system is illustrated in Fig.1. The system consists of $s$ independent subsystems and the maximal number of components hold for $i$th subsystem is $n_i$. A subsystem $i$ can work properly if at least one of its components is operational, while for each subsystem, more than one components may also work in parallel. Our investigation in this paper adopts four assumptions which were commonly adopted in most studies in this field [6], while the notations of redundancy allocation problem on parallel-series systems are presented in Table I.

- **Assumption 1:** The system consists of some subsystems, each of which can work properly if at least one of its components is operational.
- **Assumption 2:** All components states are independent.
- **Assumption 3:** Reliability, cost and weight of each components in one subsystem are same.

TABLE I

The notations of RAP on parallel-series system

| | |
|---|---|
| $R$ | reliability of the parallel system |
| $C$ | designing cost of the system |
| $W$ | weight of the system |
| $C_c$ | upper bound of cost of the system |
| $W_c$ | upper bound of weight of the system |
| $R_c$ | upper bound of reliability of the system |
| $g_1, g_2, ..., g_m$ | the m constraints of the RAP |
| $s$ | number of subsystems |
| $ai$ | number of components selected for subsystem $i$ |
| $r_i$ | reliability of every component available for subsystem $i$ |
| $c_i$ | cost of every component available for subsystem $i$ |
| $w_i$ | weight of every component available for subsystem $i$ |
| $y_i$ | quantity of component $j$ available for subsystem $i$ |
| $k_i$ | minimum number required for subsystem $i$ |
| $n_i$ | maximal number required for subsystem $i$ |
| $\theta_i, \gamma_i$ | parameters associated with the cost and weight of component in subsystem $i$ |

- **Assumption 4:** Each constraint is an increasing function of $a_i$[1].

According to the assumptions and notations, some performance metrics (e.g., system reliability, designing cost and system weight) are shown as follow:

- The reliability of the system can be calculated by

$$R = \prod_{i=1}^{s}[1 - (1 - r_i)^{a_i}], \tag{1}$$

  where $s$ is the number of subsystems, $a_i$ is the number of components available for subsystem $i$ ($1 \le a_i \le n_i$), and $r_i$ is the reliability of each available component in subsystem $i$.
- The designing cost of the system can be calculated by

$$C = \sum_{i=1}^{s} c_i[a_i + \exp(\theta_i a_i)], \tag{2}$$

  where $c_i$ is the cost of each available component in subsystem $i$, $\theta_i$ is a constant parameter for the $i$th subsystem, and $\exp(\theta_i a_i)$ is the additional cost due to the interconnection between the parallel components.
- The weight of the system is often regarded as one constraint. We can obtain the weight of the system by

$$W = \sum_{i=1}^{s} w_i[a_i + \exp(\gamma_i a_i)], \tag{3}$$

  where $w_i$ is the weight of each available component in subsystem $i$, $\gamma_i$ is a constant parameter for the $i$th subsystem, and $\exp(\gamma_i a_i)$ is a penalty factor with respect to the interconnection between the parallel components.

[1] $a_i$ is the number of components selected for $i$th subsystem

### B. SORAP and MORAP on Parallel-series System

The SORAP is to select components for every subsystem to meet the system design constraints while the reliability of the system should be maximized or the designing cost should be minimized, i.e., the SORAP is a single-objective combinatorial optimization problem with constraints. SORAP can be formulated by {Maximize $R$ (system reliability), subject to $(g_1, g_2, \ldots, g_m)$} or {Minimize $C$ (system designing cost), subject to $(g_1, g_2, \ldots, g_m)$} , where $g_1, g_2, \ldots, g_m$ are $m$ constraints. The constraints are always associated with other system variances different with the objective variance (reliability or cost), which are determined by the requirements of the consumers. The first constraint $g_1$ is associated with the designing cost of the system ($C$) when maximizing the system reliability, while $g_1$ is associated with the system reliability ($R$) when minimizing the designing cost. The formal description of SORAP is shown as follow:
Find the optimal $a_i (1 \leq a_i \leq n_i)$, $i = 1, \ldots, s$

$$\text{Max} \quad R = \prod_{i=1}^{s} [1 - (1 - r_i)^{a_i}]$$

or

$$\text{Min} \quad C = \sum_{i=1}^{s} c_i [a_i + \exp(\theta_i a_i)]$$

Subject to

$$g_1 = C = \sum_{i=1}^{s} c_i [a_i + \exp(\theta_i a_i)] \leq C_c,$$

or

$$g_1 = R = \prod_{i=1}^{s} [1 - (1 - r_i)^{a_i}] \leq R_c$$

$$g_2 = W = \sum_{i=1}^{s} w_i [a_i + \exp(\gamma_i a_i)] \leq W_c,$$

where $R_c$, $C_c$ and $W_c$ are upper bounds of $R$, $C$ and $W$ respectively.

Traditionally, SORAP is formulated with the only goal to maximizing the system reliability or minimizing the designing cost. However, sometimes the system reliability and the designing cost of system are both seriously concerned by designers. Thus, to meet different requirements of designers, it would be desirable if the reliability and costs can be involved in the optimization process simultaneously. On the other hand, we know that getting a higher reliability means we have to pay more designing cost: Equations 1 and 2 imply that we have to use a larger $a_i$ so as to achieve a higher value of $R$ while the value of $C$ will become higher at the same time. Due to the violation between the system reliability and the designing cost, we can simultaneously consider them. In other words, we can treat the system reliability and designing cost as two objectives and formulate the Multi-Objective RAP (MORAP):

TABLE II
THE IMPORTANT DEFINITIONS FOR MOEAs

| |
|---|
| Assume there are n objectives we need to optimize: |
| **Definition 1:** Dominance: a dominates b denoted as $a \prec b$ iff $\forall i: f_i(a) \leq f_i(b) \bigwedge \exists j: f_j(a) < f_j(b); i, j = 1, \ldots n.$ |
| **Definition 2:** Non-dominated: if $a \preceq b$ and $b \preceq a$ too, Then a and b are non-dominated. |
| **Definition 3:** Pareto set : A set of non-dominated solutions. |

Find the optimal $a_i$, $i = 1, \ldots, s$

$$\text{Max} \quad R = \prod_{i=1}^{s} [1 - (1 - r_i)^{a_i}]$$

and

$$\text{Min} \quad C = \sum_{i=1}^{s} c_i [a_i + \exp(\theta_i a_i)]$$

Subject to

$$g_1 = W = \sum_{i=1}^{s} w_i [a_i + \exp(\gamma_i a_i)] \leq W_c,$$

where the constraint of the MORAP is related to the weight of the system ($W$), and $W_c$ is the upper bound of $W$. To cope with MORAP, we can utilize some multi-objective approaches, such as MOEAs. In the next section, we will introduce the MOEA employed in this paper.

## III. Solving MORAP by NSGA-II

### A. Introduction of NSGA-II

During the past twenty years, evolutionary algorithms have been widely adopted in the multi-objective optimization, while the researchers have formulated a lot of efficient multi-objective optimization algorithms such as the improvement of Strength Pareto Evolutionary Algorithm SPEA2 [26], Pareto Archived Evolution Strategy (PAES) [27], Non-dominated Sorting Genetic Algorithm II (NSGA-II) [25] and so on. As a well-known MOEA, the NSGA-II is the most widely used and has been proven to perform well on various real-world application problems [28]. The pseudo-code of NSGA-II is presented in Algorithm 1.

We will employ NSGA-II in our investigation, since there have been many investigations ensuring that NSGA-II can often converge to Pareto-optimal set and the obtained solutions can often spread well over the Pareto-optimal set. NSGA-II takes the fast-non-dominated-sort mechanism to ensure the well convergence which is shown in Algorithm 2. Moreover, it adopts the Density Estimation and Crowding-Comparison Operator [25] to cut the solutions which have bad distributions so as to obtain a good spread of solutions. The above merits of NSGA-II make it a promising choice of solving MORAP. For details of NSGA-II, one can refer to [25].

**Algorithm 1** The Pseudo-Code of NSGA-II

step.1: Set the parent vector $P = \phi$, the offspring vector $Q = \phi$, the collect vector $R = \phi$ and the generation number $t = 0$.
step.2: Initialize the parent vector $P_0$.
step.3:
**while** $t <$ the terminate generation number **do**
  (1) Combine the parent and offspring population via $R_t = P_t \bigcup Q_t$.
  (2) Sort all solutions of $R_t$ to get all non-dominated fronts $F$ = fast-non-dominated-sort($R_t$) where $F = (F_1, F_2, ...)$.
  (3) Set $P_{t+1} = \phi$ and $i = 1$.
  (4)
  **while** the parent population size $|P_{t+1}| + |F_i| < N$ **do**
    (a) calculate crowding-distance of $F_i$.
    (b) add the $ith$ non-dominated front $F_i$ to the parent pop $P_{t+1}$.
    (c) $i = i + 1$.
  **end while**
  (5) Sort the $F_i$ according to the crowding distance.
  (6) Fill the parent pop $P_{t+1}$ with the first $N - |P_{t+1}|$ elements of $F_i$.
  (7) Generate the offspring population to $Q_{t+1}$.
  (8) Set $t = t + 1$.
**end while**
step.4: the population in vector $P$ are the non-dominated solutions.

**Algorithm 2** The Pseudo-Code for the function: fast-non-dominated-sort(P)

step.1: For each population $p$ in the $P$, we get the solutions which $p$ dominates and save these solutions into $S_p$. We also need to calculate the $n_p$ which is the number of solutions which dominate $p$.
step.2: Find the solutions whose $n_p = 0$ and add them to the first front $F_1$.
step.3: Initialize the front counter $i = 1$.
step.4:
**while** $F_i$ is not empty **do**
  Set the temp vector $Q = \phi$.
  **for** each $p \in F_i$ **do**
    **for** each $q \in S_p$ **do**
      $n_q = n_q - 1$.
      if $n_q = 0$ then add q to the $Q$.
    **end for**
  **end for**
  $i = i + 1$ and the solutions in $Q$ compose the $F_i$.
**end while**

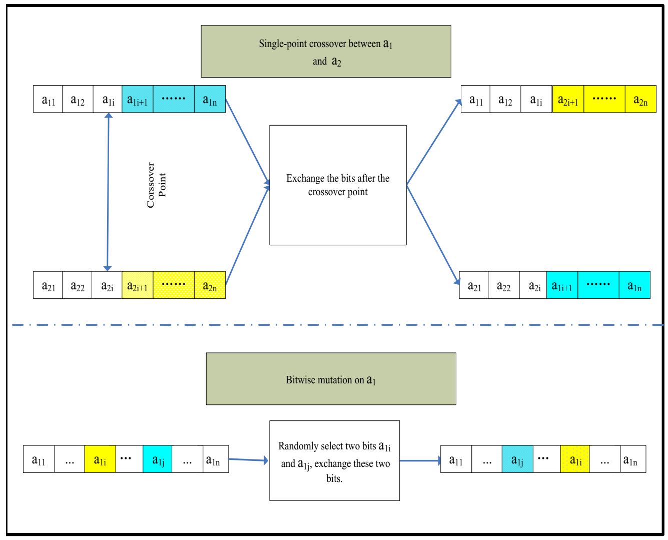


Fig. 2. The Description of Crossover and Mutation Used in This Paper

## B. Implementation Details

Although NSGA-II is a general algorithm that can be used to solve various kinds of multi-objective optimization problems, we have to utilize the problem-specific knowledge to improve NSGA-II so that it can fit different requests of real-world application problems. In this paper, on the basis of the system model and the assumptions we have introduced in the last section, we choose appropriate coding scheme and genetic operator for NSGA-II to adapt NSGA-II to MORAP.

*1) Coding Scheme:* For parallel-series systems proposed in Section II, the SORAP and MORAP are combinatorial optimization problems with the purpose of finding the number of components ($a_i$) for every subsystem, where $a_i$ is the number of components selected for $i$th subsystem with the lower bound 1 and upper bound $n_i$. Hence, we can construct a chromosome using a list of integers, and the representation of a chromosome (denoted by $p$) which has $s$ subsystems is shown as follows:

$$p = (a_1, a_2, ...., a_s), \qquad a_i \in [1, n_s],$$

where $a_i$ represents the number of components selected to the subsystem $i$.

*2) Genetic Operator:* On the basis of Coding Scheme, we adopt the single-point crossover and bitwise mutation for NSGA-II. The detailed implementation of them are presented in Fig.2. In our experiments, the crossover probability is set to be 0.9 and the mutation rate is $1/s$, where $s$ is the number of subsystems.

On the other hand, it is crucial to guide the search within the feasible region, and we have to employ some constraint handling techniques for NSGA-II. There have already been many constraints handling methods such as penalization techniques, repair techniques, separation techniques, and hybrid techniques [29]. In this paper, we utilize the constraint handling approach based on the concept of

TABLE III

PARAMETER SETTINGS OF 5-SUBSYSTEM BENCHMARK PROBLEM.

| Subsystems | $c_i$ | $r_i$ | $\theta_i$ | $w_i$ | $\gamma_i$ |
|---|---|---|---|---|---|
| 1 | 7 | 0.9 | 0.25 | 7 | 0.25 |
| 2 | 8 | 0.85 | 0.25 | 7 | 0.25 |
| 3 | 6 | 0.85 | 0.25 | 9 | 0.25 |
| 4 | 8 | 0.8 | 0.25 | 8 | 0.25 |
| 5 | 4 | 0.85 | 0.25 | 6 | 0.25 |

TABLE IV

PARAMETER SETTINGS OF 7-SUBSYSTEM BENCHMARK PROBLEM.

| Subsystems | $c_i$ | $r_i$ | $\theta_i$ | $w_i$ | $\gamma_i$ |
|---|---|---|---|---|---|
| 1 | 7 | 0.9 | 0.25 | 7 | 0.25 |
| 2 | 8 | 0.85 | 0.25 | 7 | 0.25 |
| 3 | 6 | 0.85 | 0.25 | 9 | 0.25 |
| 4 | 8 | 0.8 | 0.25 | 8 | 0.25 |
| 5 | 4 | 0.85 | 0.25 | 6 | 0.25 |
| 6 | 6 | 0.85 | 0.25 | 9 | 0.25 |
| 7 | 9 | 0.85 | 0.25 | 7 | 0.25 |

TABLE V

CONSTRAINTS OF TWO BENCHMARK PROBLEMS.

| | MORAP | SORAP |
|---|---|---|
| Example1 | $W_s = 200$ | $C_s = 180$ |
| | | $W_s = 200$ |
| Example2 | $W_s = 280$ | $C_s = 250$ |
| | | $W_s = 280$ |

constrained-dominate proposed in [25]. Concretely, a solution $i$ constrained-dominates $j$ must satisfy one of the following three conditions.

- Solution $i$ is feasible but solution $j$ is not.
- Solution $i$ and $j$ are both feasible, and $i$ dominates $j$.
- Solution $i$ and $j$ are both infeasible, but $i$ violates less constraints than $j$.

On the basis of the above concept, we can naturally transform the traditional concept of domination to the concept of constrained-domination to deal with the constraints in MORAPs.

## IV. EXPERIMENTS

In this section, we experimentally compare NSGA-II with traditional single-objective approaches on two benchmark RAPs. Two commonly used methods are chosen in our comparison. The first algorithm, named K-Y algorithm, is devised by Kim and Yum [5]. K-Y is a heuristic algorithm with the excursions over a bounded infeasible regions which can alleviate the risks of being trapped at a local optimum. The second algorithm, named R-M algorithm, is proposed by Ravi and Murty *et al.* [10]. R-M is a fast convergence algorithm which also uses an exponential cooling schedule to provides a stable global solution. These two algorithms are very classical in the reliability optimization literatures.

In the experiments, we choose two parallel-series systems as benchmark problems. One of them is with five subsystems ($s = 5$) which has been studied in [10], the other is more complex and it is with seven subsystems ($s = 7$). For the two benchmark systems, each subsystem can hold at most 6 components (i.e. $n_i = 6$ for all the subsystems, where $n_i$ represents the number of components that the $i$th subsystem can hold at most). Parameter settings of the two systems are shown in Tables III, IV and V. To compare comfortably with the K-Y and R-M algorithms, the parameter settings of the 5-subsystem benchmark problem are very similar to the parameter settings in [10]. For the 7-subsystem benchmark problem, we set the values of parameters of every subsystems similar to the 5-subsystem problem and also do a proportional increase of $C_c$ and $W_c$ according to the values of 5-subsystem problem.

### *A. Results on 5-subsystem benchmark problem*

On the basis of Tables III and V, we formulate the MORAP and SORAP on this benchmark problem in Table VI.

When we use NSGA-II to solve this MORAP, the parameter settings of NSGA-II are as follows: the crossover probability is 0.9, the mutation probability is 0.2 (i.e. $\frac{1}{s}$, here $s = 5$ ), and the terminate generation is set to 100. Moreover, we let the population size of NSGA-II be 50.

In this paper, when we use the K-Y and R-M algorithms to solve the SORAP on this problem, the cost constraint $C_c$ is set to 180. For K-Y, the maximum global iterations is 1000 and the cooling parameter is set to be 0.06.

In our experiments, all three algorithms are run on this benchmark problem for 30 times. To compare the NSGA-II with the single-objective algorithms (K-Y and R-M algorithms), we select one solution set from the 30 Pareto fronts with the median value of the diversity metric commonly used in the domain of multi-objective optimization, where the diversity of a Pareto front is given by the following formulation:

$$\Delta = \frac{\sum_{m=1}^{M} d_m^e + \sum_{i=1}^{N-1} |d_i - \overline{d}|}{\sum_{m=1}^{M} d_m^e + (N-1)\overline{d}},$$

where $d_m^e$ is the Euclidean distance between the extreme solutions of the obtained solutions and the boundary solutions of the actual Pareto set, the parameter $d_i$ is the Euclidean distance between the neighboring obtained solutions. A smaller value of $\Delta$ demonstrates a better performance of MOEA in general. On the other hand, we also record the solutions obtained by K-Y and R-M, and the solution with respect to

TABLE VI

5-SUBSYSTEM BENCHMARK PROBLEM.

| | |
|---|---|
| MORAP: | Maximize |
| | $R = \prod_{i=1}^{5} [1 - (1 - r_i)^{a_i}]$ |
| | Minimize |
| | $C = \sum_{i=1}^{5} c_i [a_i + \exp(0.25 a_i)]$ |
| | Subject to |
| | $g_1 = W = \sum_{i=1}^{5} w_i [a_i * \exp(0.25 a_i)] \leq 200$ |
| SORAP: | Maximize |
| | $R = \prod_{i=1}^{5} [1 - (1 - r_i)^{a_i}]$ |
| | Subject to |
| | $g_1 = C = \sum_{i=1}^{5} c_i [a_i + \exp(0.25 a_i)] \leq 180$ |
| | $g_2 = W = \sum_{i=1}^{5} w_i [a_i * \exp(0.25 a_i)] \leq 200$ |

TABLE VII

RESULTS OBTAINED BY K-Y AND R-M ON 5-SUBSYSTEM BENCHMARK PROBLEM.

| | K-Y | R-M | NSGA-II |
|---|---|---|---|
| $a_1$ | 2 | 2 | 2 |
| $a_2$ | 3 | 3 | 3 |
| $a_3$ | 3 | 3 | 3 |
| $a_4$ | 2 | 2 | 2 |
| $a_5$ | 3 | 3 | 3 |
| R | 0.9702 | 0.9702 | 0.9702 |
| C | 146.8368 | 146.8368 | 146.8368 |

TABLE VIII

7-SUBSYSTEM BENCHMARK PROBLEM.

| | |
|---|---|
| MORAP: | Maximize<br>$R = \prod_{i=1}^{7}[1-(1-r_i)^{a_i}]$<br>Minimize<br>$C = \sum_{i=1}^{7} c_i[a_i + \exp(0.25a_i)]$<br>Subject to<br>$g_1 = W = \sum_{i=1}^{7} w_i[a_i * \exp(0.25a_i)] \leq 280$ |
| SORAP: | Maximize<br>$R = \prod_{i=1}^{7}[1-(1-r_i)^{a_i}]$<br>Subject to<br>$g_1 = C = \sum_{i=1}^{7} c_i[a_i + \exp(0.25a_i)] \leq 250$<br>$g_2 = W = \sum_{i=1}^{7} w_i[a_i * \exp(0.25a_i)] \leq 280$ |

highest system reliability obtained by NSGA-II in Table VII respectively. Fig.3 illustrates the solutions obtained by the three algorithms, where the x-axis represents the designing cost spent by the solution while the y-axis represents the system reliability obtained by the solution,.

According to Fig.3, we find that NSGA-II has got a set of solutions which are non-dominated with each other. From Table VII and Fig.3, we also find that the solutions obtained by K-Y and R-M locate on the top of Pareto-front obtained by NSGA-II. Moreover, it can be observed that the reliability of solutions in the rectangle varies smoothly and the cost of them varies acutely. For instance, the reliability of solution A is little worse than that of the solutions obtained by K-Y and R-M algorithms, while the cost of A are very lower than that of K-Y and R-M algorithms. Thus the system designers can choose a solution in the rectangle by considering the reliability and cost simultaneously. An intuitive explanation is that K-Y and R-M algorithms maximize the system reliability while they does not concern the designing costs. Hence, they will exploit the system costs to the full extent.

### *B. Results on 7-subsystem benchmark problem*

To confirm the observation on the former benchmark problem, we further increase the maximal number of subsystem by 2. On the basis of Tables IV and V, we formulate the MORAP and SORAP on this system in Table VIII: The parameter settings of NSGA-II are the same to the settings for the former problem except that the population size has been increased to 100 here. Moreover, the terminate generation of NSGA-II is set to 150. The parameter settings of K-Y and R-M are the same to the settings for the former problem except that the cost constraint $C_c$ is set to be 250 for SORAP (due to the augment of the maximal number of subsystems). All the algorithms are repeated for 30 times. For K-Y and R-M, we select the best solutions from their 30 simulation runs.

TABLE IX

RESULTS OBTAINED BY K-Y, R-M AND THE SOLUTION WITH THE HIGHEST RELIABILITY OBTAINED BY NSGA-II ON 7-SUBSYSTEM BENCHMARK PROBLEM.

| | K-Y | R-M | NSGA-II |
|---|---|---|---|
| $a_1$ | 2 | 2 | 2 |
| $a_2$ | 3 | 3 | 3 |
| $a_3$ | 3 | 3 | 3 |
| $a_4$ | 3 | 3 | 3 |
| $a_5$ | 3 | 3 | 3 |
| $a_6$ | 3 | 3 | 3 |
| $a_7$ | 3 | 3 | 3 |
| R | 0.9656 | 0.9656 | 0.9656 |
| C | 235.338 | 235.338 | 235.338 |

The solutions obtained by K-Y, R-M and the solution with the highest reliability obtained by NSGA-II are summarized in Table IX, and all solutions obtained by the three algorithms are illustrated in Fig.4.

According to Fig.4, the solutions obtained by K-Y and R-M for SORAP, as we have observed on the last example, locate at the top of the Pareto-front obtained by NSGA-II. On the other hand, we find that NSGA-II has obtained a number of solutions which are near the solutions obtained by K-Y and R-M (i.e. the solutions in the rectangle) . The system reliability obtained by these solutions are only slightly worse than the solution given by K-Y and R-M, while the former solutions are with designing costs smaller than the latter solutions. The above facts show that MOEAs are capable of finding more promising solutions than those single-objective approaches. Such a merit will greatly increase the number of available choices for system designers.

## V. CONCLUSION

In this paper, we formulate a multi-objective combinatorial redundancy allocation problem on parallel-series modular system, and utilize NSGA-II to solve the formulated MORAP. The experiments show that NSGA-II can find a number of promising solutions of RAP. In comparison with the two widely used single-objective algorithms, NSGA-II offers much more choices for system designers. In a practical point of view, NSGA-II enables the trade-off between the system reliability and designing cost while the two widely used single-objective approaches tend to optimize the system reliability without saving any designing cost.

Although we have focused on the parallel-series systems in this paper, the general idea presented here could also be applicable to many other systems such as star structure system, circular structure system and so on. The investigations on these different systems will be carried out in our future work.

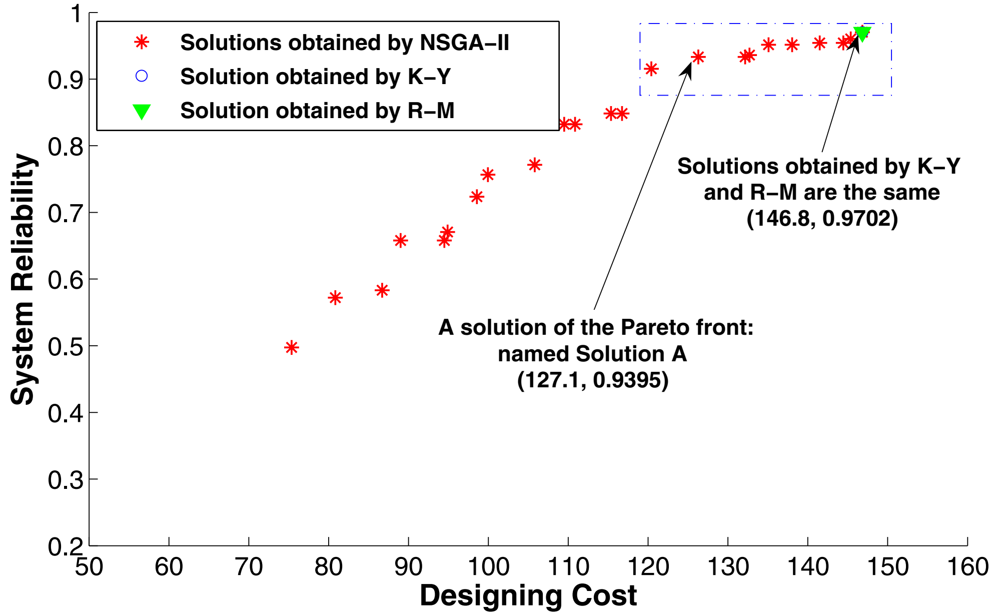


Fig. 3. Results obtained by K-Y, R-M and the solution with the highest reliability obtained by NSGA-II on 5-subsystem benchmark problem.

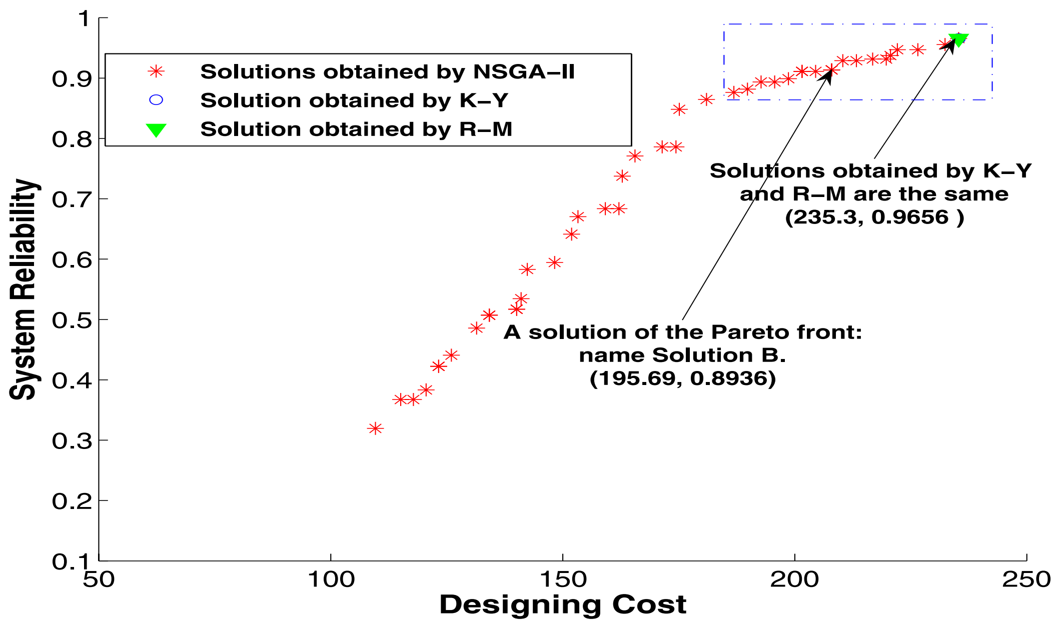


Fig. 4. Solutions obtained by NSGA-II, K-Y and R-M on the 7-subsystem benchmark problem.

## ACKNOWLEDGMENT

This work is partially supported by the Fund for Foreign Scholars in University Research and Teaching Programs (Grant No. B07033), the Fund for International Joint Research Program of Anhui Science and Technology Department (No. 08080703016), and an EPSRC grant (EP/D052785/1) on "SEBASE: Software Engineering By Automated SEarch."

## REFERENCES

[1] K. B. Misra and M. D. Ljubojecvic. "Optimal Reliability Design of A System: A New Look," *IEEE Transactions on Reliability*, vol. 22, No. 5, pp. 255–258, 1973.

[2] M. S. Chern, "On the computational complexity of reliability redundancy allocation in a series system," *Operations Research Letters*, vol. 11, pp. 309–315, 1992.

[3] D. W. Coit and A. E. Smith, "Reliability optimization of series-parallel systems using a genetic algorithm," *IEEE Transactions on Reliability*, vol. 45, no. 2, pp. 254–260, 1996.

[4] F. A. Tillman, C. L. Huang, and W. Kuo. "Determining Component Reliability and Redundancy for Optimum System Reliability," *IEEE Transactions on Reliability*, vol. 26, no. 3, pp. 162–165, 1977.

[5] J. H. Kim and B. J. Yum, “A heuristic method for solving redundancy optimization problems in complex systems,” *IEEE Transactions on Reliability*, vol. 42, no. 4, pp. 572–578, 1993.

[6] W. Kuo and V. R. Prasad, “An annotated overview of system-reliability optimization,” *IEEE Transactions on Reliability*, vol. 49, no. 2, pp. 176–187, 2000.

[7] W. Kuo and R. Wan, “Recent advances in optimal reliability allocation,” *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, vol. 37, pp. 143–156, 2007.

[8] D. L. Deeter and A. E. Smith, “Economic design of reliable network,” *IEEE Transactions on Reliability*, vol. 30, pp. 1161–1174, 1998.

[9] D. W. Coit and A. Smith, “Reliability optimization of series-parallel systems using a genetic algorithm,” *IEEE Transactions on Reliability*, vol. 45, no. 2, pp. 254–260, 1996.

[10] V. Ravi, B. S. N. Murty, and P. J. Reddy. “Nonequilibrium simulated annealing- algorithm applied to reliability optimization of complex systems,” *IEEE Transactions on Reliability*, vol. 46, pp. 233–239, 1997.

[11] J. E. Ramirez-Marquez, D. W. Coit, and A. Konak, “Redundancy allocation for series-parallel systems using a max-min approach,” *IIE Transactions*, vol. 42, no. 21, pp. 891–898, 2004.

[12] T. Yokota, M. Gen and Y. X. Li, “Genetic algorithm for nonlinear mixed integer programming problems and its applications,” *Computers and Industrial Engineering*, vol. 30, no. 4, pp. 905–917, 1996.

[13] Y. C. Liang and A. E. Smith, “An ant colony optimization algorithm for the redundancy allocation problem,” *IEEE Transactions on Reliability*, vol. 53, no. 3, pp. 417–423, 2004.

[14] J. Onishi, S. Kimura, R. J. W. James and Y. Nakagawa, “Solving the redundancy allocation problem with a mix of components using the improved surrogate constraint method,” *IEEE Transactions on Reliability*, vol. 56, no. 1, pp. 94–101, 2007.

[15] A. Billionnet, “Redundancy Allocation for Series-Parallel Systems Using Integer Linear Programming,” *IEEE Transactions on Reliability*, vol. 57, no. 3, pp. 507–516, 2008.

[16] A. K. Dhingra, “Optimal apportionment of reliability and redundancy in series systems under multiple objectives,” *IEEE Transactions on Reliability*, vol. 41, pp. 576–582, 1992.

[17] K. B. Misra, “An algorithm to solve integer programming problems: An efficient tool for reliability design,” *Microelectronics and Reliability*, vol. 31, no. 2/3, pp. 285–294, 1991.

[18] K. B. Misra and U. Sharma, “An efficient algorithm to solve integer programming problems arising in system reliability design,” *IEEE Transactions on Reliability*, vol. 40, no. 1, pp. 81–91, 1991.

[19] M. Sasaki and M. Gen, “A method of fuzzy multi-objective nonlinear programming with GUB structure by hybrid genetic algorithm,” *International Journal of Smart Engineering Design*, vol. 5, no. 4, pp. 281–288, 2003.

[20] M. Sasaki and M. Gen, “Fuzzy multiple objective optimal system design by hybrid genetic algorithm,” *Applications of Software Computation*, vol. 2, no. 3, pp. 189–196, 2003.

[21] E. P. Zafiropoulos and E. N. Dialynas, “Reliability and cost optimization of electronic devices considering the component failure rate uncertainty,” *Reliability of Engineering and System Safelty*, vol. 84, no. 3, pp. 271–284, 2004.

[22] H. A. Taboada, J. F. Espiritu and D. W. Coit, “MOMS-GA: A Multi-Objective Multi-State Genetic Algorithm for System Reliability Optimization Design Problems,” *IEEE Transcations on Reliability*, vol. 57, no. 1, pp. 182–191, 2008.

[23] S. K. Honak, D. W. Coit and F. Baheranwala, “Pruned Pareto-optimal Sets for the System Redundancy Allocation Problem Based on Multiple Prioritized Objectives”, *Journal of Heuristic*, vol. 14, no. 4, pp. 335–357, 2008.

[24] D. Salazar , C. M. Rocco and B. J. Galvn, “Optimization of Constrained Multiple-objective Reliability Problems Using Evolutionary Algorithms”, *Reliability Engineering and System Safety*, vol. 91, no. 9, pp. 1057–1070, 2006.

[25] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[26] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the strength Pareto evolutionary algorithm,” In K. Giannakoglou et al., edit, *EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pp. 95–100, Athens, Greece, 2002.

[27] J. D. Knowles and D. W. Corne, “Approximating the nondominated front using the Pareto archived evolution strategy,” *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, 2000.

[28] C. A. Coello Coello, “Evolutionary multi-objective optimisation: an historical view of the field,” *Computational Intelligence Magazine*, no.1, pp. 28–36, 2006.

[29] C. A. Coello Coello. “Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art.” *Comput Methods Appl Mech Eng* , vol. 8, no. 2, pp. 1245–1287, 2002.