A Quantum-Inspired Evolutionary Approach for non-Homogeneous Redundancy Allocation in Series- Parallel Multi-State Systems

Yan-Fu Li, Enrico Zio

A Quantum-Inspired Evolutionary Approach for non-Homogeneous Redundancy Allocation in Series-Parallel Multi-State Systems

Yan-Fu Li, Enrico Zio
Chair on Systems Science and the Energetic Challenge, European Foundation for New Energy-Electricite' de France
Ecole Centrale Paris- Supelec
Paris, France
yanfu.li@ecp.fr, yanfu.li@supelec.fr

Abstract— **Redundancy allocation is a family of well-known reliability optimization problems. The non-homogeneous type of redundancy allocation in series-parallel multi-state systems is among the most difficult ones. Evolutionary algorithms (EAs) are frequently applied to solve the problem, mainly due to the huge search space and the non-closed-form system reliability. This work proposes an efficient approach that combines a quantum-inspired evolutionary algorithm (QEA) with a newly designed local search strategy. Different from the existing EAs, it is able to evolve an explicit probabilistic model to explore the search space in an iterative way. The proposed method is tested on two benchmark problems with the comparisons to the published results. The results are promising in terms of both solution quality and computation efficiency.**

Keywords--parallel multi-state system, redundancy allocation problem, quantum inspired evolutionary algorithm, local search.

ACRONYM

LS	local search
MSS	Multi-State System
GA	Genetic Algorithm
QEA	quantum inspired evolutionary algorithm
RAP	redundancy allocation problem
SPMSS	series-parallel multi-state system
UGF	universal generating function

NOTATION

m_{ij}	highest state of component version *j* at subsystem *i*
G_i	performance variable of component *i*
$g_{i,j}$	performance level of component *i* at its state *j*
n_s	number of subsystems of MSS
C	total cost of MSS
c_{ij}	cost of component version *j* used in subsystem *i*
G_{SYS}	performance variable of MSS
$\varphi(\cdot)$	system structure function of MSS
W_{I_t}	demand presented to MSS at the *t*-th operation time interval
d_j	system adequacy level at state *j*
$A(w)$	availability function of MSS given *w*
$\|\psi\rangle$	one Q-bit
l	number of Q-bits in one Q-bit individual
n_p	number of Q-bit individual in the population
$\Delta\theta_j$	rotation angle of the *j*-th Q-bit
P_c	crossover probability of GA
P_m	mutation probability of GA

I. INTRODUCTION

Redundancy allocation problem (RAP) is a well-known optimization problem for the design of many industrial systems [1-3]. It aims to maximize system reliability or minimize system cost for given constraints on cost, reliability, weights, etc. RAP is a NP-hard [4] problem of non-linear and combinatorial nature. Most of the existing RAP works are based upon a binary state system model, which assumes that the system and its elements have only two states: perfect functioning and complete failure.

The multi-state system (MSS) model has recently gained increasing popularity for system reliability assessment because it realistically considers more than one intermediate states for the system and its elements, between the two extremes of perfect functioning and complete failure. The MSS version of the RAP has been first investigated in [5], where the universal generating function (UGF) approach [6] was used for reliability computation. Due to the high complexity of this problem, meta-heuristics are mainly used as solution techniques. The existing studies include genetic algorithm (GA) [1, 2, 7], Tabu search (TS) [8, 9], ant colony optimization (ACO) [10, 11], particle swarm optimization (PSO) [12], etc. In these implementations, the RAP is set to obtain the optimal series-parallel MSS (SPMSS) structure that minimizes the system cost while maintaining the system reliability above a predefined level.

There are two kinds of RAPs. The first kind allows only one type of component that can be used in each subsystem, namely the homogeneous RAP. The second kind allows the mixture of components in each subsystem, namely the non-

homogeneous RAP. The latter one is more challenging due to its larger solution space [13].

In this paper we propose a novel quantum-inspired evolutionary algorithm (QEA) to solve the SPMSS RAP of second type. QEA developed by Han and Kim [14] is by far the most promising application of the quantum mechanics concepts [15] onto heuristic optimization. A number of successful applications have been reported across various optimization problems [14, 16]. Based upon the concepts and principles of quantum computing, e.g. quantum bits (Q-bits), quantum gates (Q-gate) and superposition of states [17, 18], QEA is able to automatically achieve a good balance between exploration and exploitation of the solution space, and obtain quality solutions with a small population compared to the conventional evolutionary algorithms (EAs) [19].

The rest of this paper is organized as follows. The formulation of the non-homogeneous SPMSS RAP is presented in Section 2. Section 3 presents the proposed QEA approach including a novel local search (LS) strategy and constraints handling. In Section 4, the effectiveness of the proposed method is demonstrated on two benchmark problems with comparisons to published results. Section 5 concludes the work.

II. Formulation of RAP

A. Definitions and assumptions of multi-state series parallel system

The SPMSS typically consists of N subsystems connected in series. The i-th ($1 \le i \le N$) subsystem has n_i components connected in parallel, belonging to v_i versions. The j–th ($1 \le j \le v_i$) version component at the i-th subsystem has $m_{ij}+1$ states $\{0,1,\dots,m_{ij}\}$, where state m_{ij} and 0 are perfect functioning and complete failure states, respectively. The k-th ($0 \le k \le m_{ij}$) state is characterized by the performance level g_{ijk} and the state probability p_{ijk}. Additionally, there is a cost c_{ij} for the j–th version component at the i-th subsystem.

The following assumptions are made for SPMSS model:

1. The element states are mutually s-independent.
2. The mixing of components of different versions is allowed.
3. The state of the system is completely determined by the state of its components.
4. All components are repairable.

B. Formulation of RAP

Let x_{ij} denote the number of components (integer value) of j–th version at the i-th subsystem. The RAP aims to minimize the total system cost $C = \sum_{i=1}^{N}\sum_{j=1}^{v_i} c_{ij}$ while keeping the system availability A equal to or above a predefined level A_0. The formulation is presented as follows:

$$\text{Minimize:} \quad C = \sum_{i=1}^{N}\sum_{j=1}^{v_i} c_{ij}\, x_{ij} \tag{1}$$

$$\text{Subject to:} \quad A \ge A_0 \tag{2}$$

$$MAX_{ij} \ge x_{ij} \ge 0 \tag{3}$$

The second constraint specifies the range for the number of components of each version. The computation of C is straightforward. To compute A, the UGF approach is typically adopted [1, 7, 8, 12]. More details about this technique can be found in [20]. We present the basic steps in the following.

The UGF of the j–th ($1 \le j \le v_i$) version component at the i-th subsystem is

$$u_{ij}(z) = \sum_{k=0}^{m_{ij}} p_{ijk}\, z^{g_{ijk}} \tag{4}$$

The UGF of the i-th subsystem is written as

$$u_i(z) = \underbrace{u_{i1}(z)\otimes_+ \dots u_{i1}(z)}_{x_{i1}} \otimes_+ \dots \underbrace{u_{iv_i}(z)\otimes_+ \dots u_{iv_i}(z)}_{x_{iv_i}} \tag{5}$$

where the composition operator $\otimes_+$ is used to derive the UGF of a subsystem consisting of components connected in parallel. The generic composition operator $\otimes_f$ between any two combined components is defined as follows

$$u_{ij}(z)\otimes_f u_{ij^*}(z) = \sum_{k=0}^{m_{ij}}\sum_{k^*=0}^{m_{ij^*}} p_{ijk}\, p_{ij^*k^*} z^{f(g_{ijk},\, g_{ij^*k^*})} \tag{6}$$

where $f(\cdot)$ is the structure function reflecting the topology of the component combination. For examples, '+' represents the parallel combination and 'min' represents the series combination. More details about the generic composition operator $\otimes_f$ can be found in [20].

Based upon (5), the UGF of SPMSS can be written as

$$u_s(z) = u_1(z)\otimes_{min} \dots u_N(z) \tag{7}$$

Suppose it has the following expanded form

$$u_s(z) = \sum_{i^*=1}^{N^*} p_{i^*} z^{g_{i^*}} \tag{8}$$

Given the arbitrary system demand W_{I_t} at the t-th operation time step, the system availability A_{I_t} at this time is computed as

$$A_{I_t} = \Psi\left(\sum_{i^*=1}^{N^*} p_{i^*} Z^{g_{i^*}-W_{I_t}}\right) = \sum_{i^*=1}^{N^*} \Psi\left(p_{i^*} Z^{g_{i^*}-W_{I_t}}\right) \tag{9}$$

where Ψ is the distributive operator [20] with the following definition

$$\Psi(pZ^{g-w}) = \begin{cases} p, & if\ g \ge W \\ 0, & if\ g < W \end{cases} \tag{10}$$

For the entire operation period which is divided into T time steps, the system availability A is computed as

$$A = \left(\sum_{I_t=1}^{T} A_{I_t}\, d_{I_t}\right) \Big/ \sum_{I_t=1}^{T} d_{I_t} \tag{11}$$

where d_t is the duration of the t-th time step.

III. QEA Approach

A. Solution representation

In analogy to the bit in conventional EA encoding, the Q-bit, i.e. quantum bit [21], serves as the smallest information unit in QEA. Unlike the classical bit, which has to be either

state '0' or state '1', a Q-bit can be '0', '1', or a superposition of both states. Let $|0\rangle$ and $|1\rangle$ denote the two basis states, respectively; the state of one Q-bit $\begin{bmatrix}\alpha\\ \beta\end{bmatrix}$ can be written as

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{12}$$

where α and β are the probability amplitudes and they have to satisfy that

$$|\alpha|^2 + |\beta|^2 = 1 \tag{13}$$

It is noted that $|\alpha|^2$ and $|\beta|^2$ are the probabilities of state '0' and state '1', respectively. A Q-bit individual is a string of l concatenated Q-bits

$$\boldsymbol{q} = [q_1\ q_2\ \dots\ q_l] = \begin{bmatrix}\alpha_1 & \alpha_2 & \dots & \alpha_l\\ \beta_1 & \beta_2 & \dots & \beta_l\end{bmatrix} \tag{14}$$

For each Q-bit q_i, the condition (13) must be satisfied. To evaluate the fitness of a Q-bit individual $\boldsymbol{q}$, each q_i is first sampled to form a binary bit $b_i \in \{0, 1\}$. This sampling is done according to the probability $|\beta_i|^2$ of state '1'. In fact, $\boldsymbol{q}$ defines a probabilistic model

$$\boldsymbol{p} = [|\beta_1|^2\ \ |\beta_2|^2\ \dots\ |\beta_l|^2] \tag{15}$$

This model explicitly describes the probability distribution of the solutions in the search space and is able to sample 2^l different binary bit solutions.

B. QEA procedures

The detailed procedures of QEA are presented as follows.

Initialization: set t, the generation index, equal to 0 and randomly generate the population $\boldsymbol{Q}_t = \{\boldsymbol{q}_1^t, \dots, \boldsymbol{q}_{n_p}^t\}$ (where n_p is the total number of individuals in the population). Each individual $\boldsymbol{q}_i^t$ $(i = 1, \dots, n_p)$ takes the form as presented in eq. (14) and all its Q-bits $q_{ij}^t = \begin{bmatrix}\alpha_{ij}^t\\ \beta_{ij}^t\end{bmatrix}$ $(j = 1, \dots, l)$ equal to the value $1/\sqrt{2}$ so that the probabilities of observing $|1\rangle$ and $|0\rangle$ are the same for each Q-bit.

Observation*:* sample a binary population $\boldsymbol{B}_t = \{\boldsymbol{b}_1^t, \dots, \boldsymbol{b}_{n_p}^t\}$ from $\boldsymbol{Q}_t$. For each individual $\boldsymbol{b}_i^t = [b_{i1}^t, \dots, b_{il}^t]$, each of its element b_{ij}^t is binary and determined by comparing $\left|\beta_{ij}^t\right|^2$ with a uniformly distributed random number in the range [0, 1]. If $\left|\beta_{ij}^t\right|^2 > rand[0, 1]$ then $b_{ij}^t = 1$; otherwise $b_{ij}^t = 0$.

Evaluation: evaluate each individual in $\boldsymbol{B}_t$ using the fitness or objective function. In this study, the fitness function is a penalized form of the system cost (1). (see after its definition (17))

Elitism: create a population of elite solutions $\boldsymbol{E}_t = \{\boldsymbol{e}_1^t, \dots, \boldsymbol{e}_{n_p}^t\}$ to store each binary individual $\boldsymbol{b}_i^t$ initially sampled. It is noted that $\boldsymbol{E}_t$ can be divided into a number of equally sized *local groups*. Within each group, the solutions have the ability to synchronize themselves with the best individual among them, periodically. In addition, all the solutions in $\boldsymbol{E}_t$ are periodically replaced by the best one $\boldsymbol{e}_t$ found in the entire $\boldsymbol{E}_t$. More details about the elitism strategy (also named 'migration' by Han and Kim) can be found in [14].

Set $t = t + 1$.

Observation: sample a binary population $\boldsymbol{B}_t$ from $\boldsymbol{Q}_{t-1}$.

Evaluation: evaluate each individual in $\boldsymbol{B}_t$ using the fitness or objective function $f(\cdot)$.

Variation: update each Q-bit individual using the Q-gate [14], which is the analog to variation operators such as crossover and mutation in classical EA. In QEA, the variation operator is the rotation gate $U(\Delta\theta_j)$,

$$U(\Delta\theta_j) = \begin{bmatrix}\cos(\Delta\theta_j) & -\sin(\Delta\theta_j)\\ \sin(\Delta\theta_j) & \cos(\Delta\theta_j)\end{bmatrix} \tag{15}$$

where $\Delta\theta_j$ is the rotation angle determining the magnitude and direction of the rotation for the j-th Q-bit, and it should be designed in compliance with the application problem. Using this gate, the j-th Q-bit of $\boldsymbol{q}_i^{t-1}$ is updated as follows,

$$\begin{bmatrix}\alpha_{ij}^t\\ \beta_{ij}^t\end{bmatrix} = U(\Delta\theta_j)\begin{bmatrix}\alpha_{ij}^{t-1}\\ \beta_{ij}^{t-1}\end{bmatrix} \tag{16}$$

It is seen that this operator should satisfy the normalization condition. Figure 1 shows the polar plot of such updating.

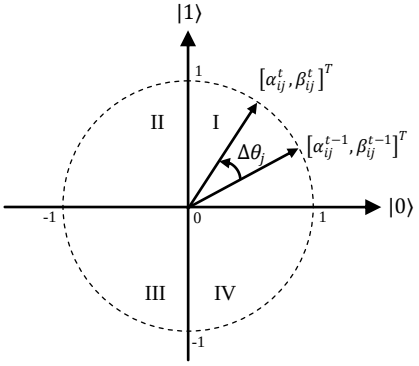

Figure 1. Polar plot of the rotation gate for updating one Q-bit

Prior to each updating, $\Delta\theta_j$ has to be determined. Table 1 summarizes the rules for $\Delta\theta_j$ value assignments considering minimizing the fitness function $f(\cdot)$. Note that the same table can be used for maximization problems. To explain this Table we take its first row, $f(\boldsymbol{b}_i^t) > f(\boldsymbol{e}_i^t)$, and $b_{ij}^t = 0$ and $e_{ij}^t = 1$, where $\Delta\theta_j$ is determined as follows: 1) if the Q-bit q_{ij}^t is in the III or I quadrant (as shown in Fig 1), then $\Delta\theta_j$ will be a

positive value (or δ_θ) which indicates the increase of probability of sampling state $|1\rangle$; 2) if q_{ij}^t is in the II or IV quadrant, $\Delta\theta_j$ will be a negative value (or $-\delta_\theta$) which means an increase of the probability of sampling state $|0\rangle$.

TABLE 1. LOOKUP TABLE OF THE ROTATION ANGLE

$f(\boldsymbol{b}_i^t) > f(\boldsymbol{e}_i^t)$	b_{ij}^t	e_{ij}^t	$\Delta\theta_j$
True	0	1	δ_θ (Q-bit in I/III quadrant) $-\delta_\theta$ (Q-bit in II/IV quadrant)
	1	0	$-\delta_\theta$ (Q-bit in I/III quadrant) δ_θ (Q-bit in II/IV quadrant)
	0	0	0
	1	1	0
False	0	1	0
	1	0	0
	0	0	0
	1	1	0

Elitism: obtain $\boldsymbol{E}_t$ by assigning to each $\boldsymbol{e}_i^t$ the best individual from the pair $\boldsymbol{b}_i^t$ and $\boldsymbol{e}_i^{t-1}$.

Termination: stop the algorithm if the termination criteria are met; otherwise go to Step 6.

In literature there are a number of Q-gates, e.g. NOT gate, controlled NOT gate, or Hadamard gate [21]. The rotation gate is most frequently applied in QEA.

C. QEA-SPMSS-RAP Approach

In this Section, the classical QEA is tailored for SPMSS RAP. In particular, it includes the encodings of the solution to RAP, the constraints handling, the novel local search strategy, and the complete procedures of the QEA approach.

Encodings

In line with the problem formulation in Section 2.2, a solution to RAP can be represented by a vector $\boldsymbol{x} = [x_{11}, \dots, x_{1v_1}; \dots; x_{N1}, \dots, x_{Nv_N}]$. Because the QEA operation is based upon binary variables, for each integer variable x_{ij} within the range $[0, MAX_{ij}]$ we use its binary equivalent $[b_{ij1}, \dots, b_{ijn_{ij}}]$ where $n_{ij} = \lceil \log_2 MAX_{ij} \rceil$. Gray coding is used for the decimal-binary conversion, because in this system two successive values are different by only one bit. The Q-bit individual takes the form described in eq. (14) with the length equal to that of its corresponding binary individual.

Constraints handling

The penalty function approach is used to handle the constraint in eq. (2). To effectively explore the feasible and infeasible solutions near the border of the feasible area, a penalty approach inspired by the BSS work [22, 23] is used. It is then added onto the original system cost function. The penalized system cost as the following expression,

$$C_p = \begin{cases} C, & A \geq A_0 \\ C + d\left(1 + \frac{A_0}{A}\right), & otherwise \end{cases} \quad (17)$$

where C is the original system cost presented in eq. (1) and d is a relatively large constant dependent on the specific problem.

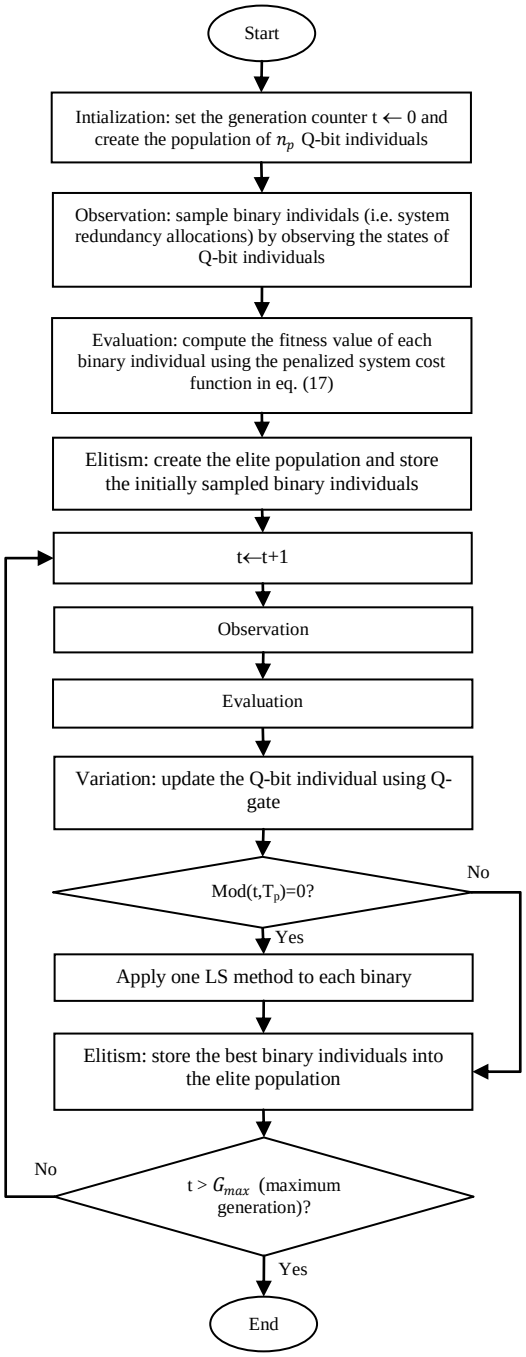

Figure 2. Flowchart of the QEA-SPMSS-RAP approach

Local search (LS)

LS has shown to be effective in improving the candidate solutions obtained by the main algorithm for solving RAP [12, 22]. In this study, we design two LS methods alternately applied to each individual in the binary population. They are modified from the LS strategies c) and e) proposed in [12], with the emphasis on exploring less expensive solutions. At each LS operation, only one of the LS methods is performed

on a randomly selected subsystem i_r; the other LS method is performed at the next LS operation.

The first LS method deletes one component of a randomly selected version $j_{r\_}$ whose $x_{i_r j_{r\_}} > 0$ (i.e. $x_{i_r j_{r\_}} \leftarrow x_{i_r j_{r\_}} - 1$); then, it adds one component to a randomly selected version j_{r_+} which is less expensive than version $j_{r\_}$ (i.e. $x_{i_r j_{r_+}} \leftarrow \min\left(x_{i_r j_{r_+}} + 1, MAX_{i_r j_{r_+}} \right)$). Note that the upper limit $MAX_{i_r j_{r_+}}$ must be kept. If the component being deleted is of the least expensive version, then no component will be added. For example, if a subsystem consists of one version 1 component of cost 35 $ and nominal performance 50%, three version 2 components of cost 20 $ and nominal performance 30%, and two version 3 components of cost 15 $ and nominal performance 25%, then a feasible LS operation is to remove one version 2 component and add one version 3 component. Another option is to simply delete one version 3 component.

The second LS method randomly selects one existing component version $j_{r\_}$ to be deleted (i.e. $x_{i_r j_{r\_}} \leftarrow 0$), and then randomly select another version $j_{r_+} (r_+ \neq r\_)$ to be increased. The number of components to be added is $n_{r_+} = round\left(x_{i_r j_{r\_}} \cdot g_{i_r j_{r\_} m_{i_r j_{r\_}}} / g_{i_r j_{r_+} m_{i_r j_{r_+}}} \right)$, where $g_{ij\,m_{ij}}$ is the nominal performance of the component version j at subsystem i. If $n_{r_+} c_{i_r j_{r_+}} < x_{i_r j_{r\_}} c_{i_r j_{r\_}}$, then $x_{i_r j_{r_+}} \leftarrow \min\left(x_{i_r j_{r_+}} + n_{r_+}, MAX_{i_r j_{r_+}} \right)$; otherwise, perform neither the deletion nor the addition. Using the exemplar system above, a feasible LS operation is to remove one version 1 component and add two version 3 components. However, to remove two version 3 components and add one version 1 component is not feasible.

Overall procedure

The overall procedure of the proposed optimization approach is represented by the flow chart in Figure 2. Note that T_p denotes the period for LS operation and it starts at the first LS method application.

IV. Empirical Validations

A. Experiment design

To compare with other published algorithms, the proposed QEA approach is tested on two well known benchmark problems. The first problem (P1) consists of four subsystems connected in series [2]. For each subsystem, there are 4 to 6 different component versions available. The availability requirement A_0 is set to three different values, namely 0.900, 0.960, 0.990, to create three test cases. The second problem (P2) consists of five subsystems connected in series [7]. For each subsystem, there are 4 to 9 different component versions available. The availability requirement is set to be 0.975, 0.980 and 0.990. The data sets of the two problems can be found in [24]. The upper limit MAX_{ij} is set to be 7, the same to all test cases.

The parameters of QEA approach include population size n_p, maximum generation G_{max}, absolute rotation angle δ_θ, penalty constant d, and LS period T_p. As QEA typically needs a very small population, we set $n_p = 5$ for all the experiments. We set G_{max} = 2000 for P1 and P2. The value of δ_θ is problem-dependent [14]. To choose an optimal one, in this study we change δ_θ from 0.005π to 0.050π with step size of 0.005π, following [16]. For each problem, different δ_θ values are first evaluated on the test case with $A_0 = 0.99$ and, then, the δ_θ value which produces the lowest average cost is used for all the test cases of this problem. In the end, we have $\delta_\theta = 0.010\pi$ and 0.030π for P1and P2, respectively. The penalty constant d needs to have a sufficiently large value [1, 2]: we set d = 100 for P1 and P2. Finally, we set T_p= 10 for all test cases, following [12]. Due to the stochastic nature of the search algorithm, the QEA approach is run 20 times for each test case. All the experiments have been carried out in MATLAB software package, on a PC with Intel Core i5 of 3.4 GHz and 4 GB RAM.

B. Results and comparisons to published results

For each test case, the best, average and worst minimal cost values of the 20 experiment runs are recorded. Table 2 summarizes these results.

TABLE 2. RESULTS FOUND BY QEA METHOD ON ALL PROBLEMS

Problem	A_0	C ($)		
		Best	Average	Worst
P1	0.990	8.180	8.355	8.555
	0.960	7.009	7.381	7.803
	0.900	5.423	5.901	6.477
P2	0.990	15.870	15.923	16.087
	0.980	14.770	14.893	15.237
	0.975	12.855	12.999	13.126

Table 3 presents the detailed information about the best solution of each test case found by QEA approach out of 20 runs. The solution is represented in the form, '$j(x_{ij})$' for a subsystem i. For example, 3(3) in the top cell of the solution column '2' indicates three type 3 components in subsystem 2. The reliability and cost values of each solution are presented as well.

TABLE 3. BEST SOLUTIONS FOUND BY QEA METHOD

	A_0	A	C ($)	Solution				
				1	2	3	4	5
P1	0.990	0.992	8.180	1(3)	3(3)	1(3)	3(1) 4(2)	
	0.960	0.963	7.009	1(3)	2(1) 3(2)	1(3)	3(1) 5(1)	
	0.900	0.901	5.423	4(1)	3(2)	1(3)	3(1) 5(1)	
P2	0.990	0.992	15.870	4(2) 6(1)	3(2)	2(2) 3(1)	7(3)	4(3)
	0.980	0.980	14.770	4(2) 6(1)	3(2)	2(1) 3(2)	7(3)	3(2) 4(1)
	0.975	0.976	12.855	4(2) 6(1)	5(6)	1(1) 4(1)	7(3)	4(3)

In Table 4, the outcomes of QEA experiments are compared with published results, in terms of best solution quality and

number of fitness evaluations. Since various algorithms were tested using different computing facilities including the hardware platforms and software packages, the number of fitness evaluations is a more reliable metric of computational efficiency compared to the actual computation time. It is seen from Table 4 that across all test cases, the proposed approach requires the lowest numbers of fitness evaluations among the methods who have achieved the best solutions (i.e. GA, SP/TG, and PSP/LS). For P2 the proposed method is at least 10 times faster than other methods. As to P1, the proposed method is about twice faster than SP/TG and about 10 times faster than PSP/LS.

Note that we use the maximum number of fitness evaluations for SP/TG, since the exact number of fitness evaluations are not shown in the paper [9]. Due to the tabu search strategies, SP/TG might need less fitness evaluations in real applications. Nevertheless, the same type of Tabu search can be incorporated into our algorithm for further reduction of computation efforts. As to ACO, it has the smallest number of evaluations for P2, but it does not obtain the best solutions.

TABLE 4. COMPARISONS TO THE PUBLISHED RESULTS

Publication		P1			P2		
	A_0	0.990	0.960	0.900	0.990	0.980	0.975
GA [7]	C ($)				**15.870**	**14.770**	**12.855**
	# of FE*				1.03e6	1.03e6	1.03e6
ACO [11]	C ($)				16.314	14.885	13.101
	# of FE				4.5e3	4.5e3	4.5e3
SP/TG [9]	C ($)	**8.180**	**7.009**	**5.423**	**15.870**	**14.770**	**12.855**
	# of FE	<2.05e4	<2.05e4	<2.05e4	<1.005e5	<1.005e5	<1.005e5
PSP/LS [12]	C ($)	**8.180**	**7.009**	**5.423**	**15.870**	**14.770**	**12.855**
	# of FE	>1.0e5	>1.0e5	>1.0e5	>2.0e5	>2.0e5	>2.0e5
QEA	C ($)	**8.180**	**7.009**	**5.423**	**15.870**	**14.770**	**12.855**
	# of FE	**1.1e4**	**1.1e4**	**1.1e4**	**1.1e4**	**1.1e4**	**1.1e4**

* FE stands for fitness evaluation

V. CONCLUSIONS AND FUTURE WORKS

In this work, we have considered the SPMSS heterogeneous RAP. QEA is first introduced as the solution method. An efficient LS strategy is originally designed to enhance the exploitation ability of QEA. The validations on 6 benchmark test cases with comparisons to published results show that the proposed QEA approach is able to achieve the best solutions using much less computation resources than other methods. Given the promising results obtained, future works can be devoted to extending the application of this method to larger size SPMSS heterogeneous RAP or more complex RAPs, e.g. networks structure optimization, RAP under random-fuzzy environments, etc.

REFERENCES

[1] A. Lisnianski, G. Levitin, H. Ben-Haim, and D. Elmakis, "Power system structure optimization subject to reliability constraints," *Electric Power Systems Research*, vol.39, no.2, pp. 145-152, 1996.

[2] G. Levitin, A. Lisnianski, H. Ben-Haim, and D. Elmakis, "Redundancy optimization for series-parallel multi-state systems," *Reliability, IEEE Transactions on*, vol.47, no.2, pp. 165-172, 1998.

[3] S. Kulturel-Konak, A. E. Smith, and D. W. Coit, "Efficiently solving the redundancy allocation problem using tabu search," *IIE transactions*, vol.35, no.6, pp. 515-526, 2003.

[4] M.-S. Chern, "On the computational complexity of reliability redundancy allocation in a series system," *Operations Research Letters*, vol.11, no.5, pp. 309-315, 1992.

[5] I. Ushakov, "Optimal standby problems and a universal generating function," *Sov J Comput Syst Sci*, vol.25, no.4, pp. 79-82, 1987.

[6] I. Ushakov, "Universal generating function," *Soviet Journal of Computer and System Sciences*, vol.24, no.5, pp. 37-49, 1986.

[7] G. Levitin, A. Lisnianski, and D. Elmakis, "Structure optimization of power system with different redundant elements," *Electric Power Systems Research*, vol.43, no.1, pp. 19-27, 1997.

[8] M. Ouzineb, M. Nourelfath, and M. Gendreau, "An efficient heuristic for reliability design optimization problems," *Computers & Operations Research*, vol.37, no.2, pp. 223-235, 2010.

[9] M. Ouzineb, M. Nourelfath, and M. Gendreau, "A heuristic method for non-homogeneous redundancy optimization of series-parallel multi-state systems," *Journal of Heuristics*, vol.17, no.1, pp. 1-22, 2011.

[10] Y. Massim, A. Zeblah, R. Meziane, M. Benguediab, and A. Ghouraf, "Optimal design and reliability evaluation of multi-state series-parallel power systems," *Nonlinear Dynamics*, vol.40, no.4, pp. 309-321, 2005.

[11] M. Nourelfath, N. Nahas, and A. Zeblah, "An ant colony approach to redundancy optimization for multi-state system," in *International conference on industrial engineering and production management (IEPM'2003), Porto-Portugal*, 2003.

[12] Y. Wang and L. Li, "Heterogeneous Redundancy Allocation for Series-Parallel Multi-State Systems Using Hybrid Particle Swarm Optimization and Local Search," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol.42, no.2, pp. 464-474, 2012.

[13] L. Zia and D. W. Coit, "Redundancy allocation for series-parallel systems using a column generation approach," *Reliability, IEEE Transactions on*, vol.59, no.4, pp. 706-717, 2010.

[14] K.-H. Han and J.-H. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," *Evolutionary Computation, IEEE Transactions on*, vol.6, no.6, pp. 580-593, 2002.

[15] M. D. Platel, S. Schliebs, and N. Kasabov, "Quantum-inspired evolutionary algorithm: a multimodel EDA," *Evolutionary Computation, IEEE Transactions on*, vol.13, no.6, pp. 1218-1232, 2009.

[16] T. Lau, C. Chung, K. Wong, T. Chung, and S. Ho, "Quantum-inspired evolutionary algorithm approach for unit commitment," *Power Systems, IEEE Transactions on*, vol.24, no.3, pp. 1503-1512, 2009.

[17] A. Narayanan and M. Moore, "Quantum-inspired genetic algorithms," in *Evolutionary Computation, 1996., Proceedings of IEEE International Conference on*, 1996, pp. 61-66.

[18] K.-H. Han and J.-H. Kim, "Genetic quantum algorithm and its application to combinatorial optimization problem," in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, 2000, vol.2, pp. 1354-1360.

[19] K.-H. Han, K.-H. Park, C.-H. Lee, and J.-H. Kim, "Parallel quantum-inspired genetic algorithm for combinatorial optimization problem," in *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, 2001, vol.2, pp. 1422-1429.

[20] G. Levitin, *The universal generating function in reliability analysis and optimization*: Springerverlag London Limited, 2005.

[21] T. Hey, "Quantum computing: an introduction," *Computing & Control Engineering Journal*, vol.10, no.3, pp. 105-112, 1999.

[22] Y.-C. Liang and A. E. Smith, "An ant colony optimization algorithm for the redundancy allocation problem (RAP)," *Reliability, IEEE Transactions on*, vol.53, no.3, pp. 417-423, 2004.

[23] D. W. Coit and A. E. Smith, "Penalty guided genetic search for reliability design optimization," *Computers & industrial engineering*, vol.30, no.4, pp. 895-904, 1996.

[24] M. Ouzineb, M. Nourelfath, and M. Gendreau, "Tabu search for the redundancy allocation problem of homogenous series–parallel multi-state systems," *Reliability Engineering & System Safety*, vol.93, no.8, pp. 1257-1272, 2008.