Hindawi Publishing Corporation
Science and Technology of Nuclear Installations
Volume 2015, Article ID 516470, 14 pages
http://dx.doi.org/10.1155/2015/516470

Research Article

An Improved Shuffled Frog Leaping Algorithm for Assembly Sequence Planning of Remote Handling Maintenance in Radioactive Environment

Jianwen Guo,[1] Hong Tang,[2] Zhenzhong Sun,[1] Song Wang,[3] Xuejun Jia,[4] Haibin Chen,[1] and Zhicong Zhang[1]

[1] *School of Mechanical Engineering, Dongguan University of Technology, Dongguan 523808, China*
[2] *Dongguan Neutron Science Center, Dongguan 523890, China*
[3] *College of Mechanical & Automotive Engineering, South China University of Technology, Guangzhou 510641, China*
[4] *Institute of High Energy Physics, Chinese Academy of Sciences, Beijing 100049, China*

Correspondence should be addressed to Jianwen Guo; guojw@dgut.edu.cn

Received 2 October 2014; Revised 13 January 2015; Accepted 28 January 2015

Academic Editor: Massimo Zucchetti

Assembly sequence planning (ASP) of remote handling maintenance in radioactive environment is a combinatorial optimization problem. This study proposes an improved shuffled frog leaping algorithm (SFLA) for the combinatorial optimization problem of ASP. An ASP experiment is conducted to verify the feasibility and stability of the improved SFLA. Simultaneously, the improved SFLA is compared with SFLA, genetic algorithm, particle swarm optimization, and adaptive mutation particle swarm optimization in terms of efficiency and capability of locating the best global assembly sequence. Experiment results show that the proposed algorithm exhibits outstanding performance in solving the ASP problem. The application of the proposed algorithm should increase the level of ASP in a radioactive environment.

1. Introduction

Radioactive installation (e.g., nuclear power plants and high-energy physics research institutes) generally has the characteristics of complicated structures, high speed, and heavy loads. The installations themselves and their working environments are radioactive. The aforementioned characteristics may cause failures of key equipment in radioactive installation, which seriously affect the lifetime of radioactive installation.

Maintenance refers to restoring aging or faulty equipment parts to a satisfactory operating condition. It includes inspection, testing, diagnosis, disassembly, assembly, cleaning, repair, and replacement. Key equipment of radioactive installation that provides base functions must be maintained during installation lifetime [1–3]. In radioactive installation, most maintenance activities are conducted in a radioactive environment that is unsuitable for humans; in such cases, remote handling maintenance (RHM) is necessary [4]. RHM enables a person to manually handle work without being physically present at a work site through a manipulator or a robot [5].

Radioactive equipment has a complex structure that causes difficulty in maintenance operations within a radioactive installation. Maintenance procedures must be planned in advance to ensure reliability and security of RHM [6]. Remote handling maintenance planning (RHMP) predetermine the maintenance procedures of radioactive equipment during the design of radioactive installation [7]. Assembly sequence planning (ASP) determines the order in which each part and subassembly must be inserted into an incrementally expanding subassembly that eventually leads to a final assembly [8]. Assembly operation is part of RHM procedure. Thus, ASP is considered as a subdomain of RHMP.

In RHM, ASP provides an optimal sequence to replace aging or faulty parts under certain constraint conditions (e.g., time, cost, and reliability). Complex radioactive equipment with a large number of parts has several feasible sequences

for assembly. Finding an optimal assembly sequence to satisfy time, cost, and reliability requirements is a combinatorial problem; the complexity of this problem is proportional to the number of equipment parts. The number of feasible RHM sequences increases with equipment complexity [9]. A large number of equipment parts results in a combination explosion, and the optimal solution is omitted. ASP is then shown to be NP-complete [10].

Optimization techniques based on principles inspired by natural systems have been proposed over the past decades to solve the combinatorial explosion problem [11]. The shuffled frog leaping algorithm (SFLA) is a recent metaheuristic optimization algorithm that is inspired by the memetic evolution of a group of frogs when seeking food. SFLA involves a set of frogs that cooperate to achieve a unified behavior for the entire system, which produces a robust system that can find high-quality solutions to problems with large search spaces [12]. SFLA exhibits global searching capability, rapid convergence, and strong robustness. It has been successfully applied to several fields. However, SFLA is suitable for continuous optimization [13]. SFLA should be improved when applied to ASP, which is a discrete search and optimization problem.

In this study, an improved SFLA is presented to solve the ASP problem for RHM in radioactive environment. Each SFLA operation is redefined. In particular, a swap operator and a swap sequence are introduced and the local search strategy is designed to directly search the discrete domain. A diversity control strategy based on genetic algorithm (GA) is proposed to improve the search for global optimal solutions. An ASP experiment show that the algorithm exhibits outstanding performance in solving the ASP problem.

The remainder of the paper is organized as follows. Section 2 introduces related works. Section 3 describes SFLA. Section 4 states the ASP problem. Section 5 discusses the improved SFLA for ASP. Section 6 describes the experiments and the analyses. Finally, Section 7 provides the conclusions of the study.

2. Related Studies

2.1. RHM. Maintenance preserves or restores a system or facility to its desired state. The following problems should be considered for maintenance in radioactive installation: (1) safety of the maintenance worker, in cases where humans cannot gain access because of the high radiation dose rate; (2) feasibility of maintenance work, in cases where humans has difficulty working with equipment because of certain conditions (e.g., small spaces and narrow gaps); and (3) reliability of maintenance work, in cases where harsh environments and heavy workloads cause human errors.

RHM is applied to solve the aforementioned problems [3–5]. RHM mainly repairs fault parts that cause equipment to stop working in a radioactive environment. Operations of RHN mainly include replacement and disposal works, which are remotely handled by using power and master-slave manipulators [14]. The following observations are made. (1) RHM differs from conventional equipment maintenance because it employs a robot or a remote operation tool in a hot cell instead of a human. (2) The robot must be teleoperated, fully controlled, or supervised by a human outside the hot cell because the majority of RHM tasks require the intuition and intelligence of a human [5]. (3) A human does not need to be physically present at the work site to conduct maintenance work.

RHM will be applied to radioactive installation such as the China Fusion Engineering Test Reactor [15], the International Thermonuclear Experimental Reactor (ITER) [6], and the European Organization for Nuclear Research [16].

2.2. Robots in RHM. Several robots have been developed for RHM. Takeda et al. [17] designed three kinds of robots that can transport different parts in a radioactive environment. The French Atomic Energy Agency Interactive Robotics Laboratory developed an industrial robot system for a nuclear spent fuel reprocessing plant [18]. The robot, which uses RX170 as a slave arm and a control platform called TAO2000 V2, supports a master-slave operation with a force feedback and tolerates radiation up to a 10 kGy integrated dose [19]. Sanders [20] developed a remote handling system with "man in the loop" approach that provides the remote robot operator for the Joint European Torus. Vale et al. [21] developed an autonomous mobile robot for ITER. Terada et al. [22] designed and developed a pick-and-place work robot to cope with the module placement for the semiconductor tracker barrel assembly. The robot can place modules with a mechanical precision of over 25 μm. Lee et al. [23] developed a bridge transported servo-manipulator system to overcome the limited workspace of conventional mechanical master-slave manipulators in a hot cell. Lee et al. [24] developed a cable-driven dual arm master-slave servo-manipulator for the pyroprocess research facility.

2.3. RHMP. In radioactive installation, RHMP predetermines the maintenance process during the stages of radioactive installation design. Traditional RHM is mainly performed through an empirical design and physical verification by experiment platform. It is unsuitable for complex structure installation and can be laborious and ineffective. The experiment platform results in high costs and a long cycle.

The development of computer, artificial intelligence, and simulation technologies allows the application of virtual maintenance planning to RHM [25]. Takeda et al. [26] developed a virtual reality simulator to support the Banket simulation of the RHM process of ITER. Heemskerk et al. [27] studied the simulation process dynamics based on the ITER RHM simulator. Geng et al. [28] developed a novel virtual maintenance application for maintenance safety evaluation to provide recommendations on maintenance safety. Esque et al. [29] completed a digital simulation model of the ITER separator. Shuff et al. [30] developed a set of discrete event simulation tools for the remote operating process planning of the ITER hot cabinet. Robbins et al. [31] achieved a real-time and visualized track of remote operating process planning by using virtual reality and an intelligent database. Park et al. [32,

33] studied visualization and simulation of a nuclear facility disassembling process and established an RHM system.

In the aforementioned studies, maintenance sequence was generally obtained in an exploratory manner because of the lack of a guided optimized maintenance sequence. Such sequence may be inefficient and the optimal solution might be omitted without intelligent support. Introducing intelligent planning technologies to ASP is essential to further enhance virtual maintenance planning [7].

2.4. Intelligent ASP for Complex Products. Assembling products (by humans or by robots) is the act of combining parts of equipment. ASP obtains the order for each part and subassembly, which is then inserted into an incrementally expanding subassembly that eventually leads to a final assembly. ASP is a combinatorial problem in complex products. Solving this problem with human involvement is difficult and impractical because of the combinatorial explosion issue. Research in intelligence assembly sequencing has rapidly increased in recent years. The intelligence ASP problem is regarded as a discrete search and optimization problem. Various artificial intelligence approaches have been proposed recently, including graph theory [34, 35], subassembly detection [36, 37], motion planning [38], and evolution algorithm [11].

Evolution algorithms provide new solutions to various complex optimization problems by imitating the self-organization mechanism of natural biological communities and the adaptive ability of evolution. Chen and Xiao [39] developed an enhancing artificial bee colony algorithm with self-adaptive searching strategy and artificial immune network operators for global optimization. Xu et al. [40] developed an improved genetic algorithm for distribution network planning. Chen and Ju [41] developed a novel artificial bee colony algorithm for solving the supply chain network design under disruption scenarios. Cheng et al. [42] developed a metaheuristics for airport gate assignment. Lorpunmance and Sap [43] developed an ant colony optimization for dynamic job scheduling in grid environment. Evolution algorithms, such as GA [44–46], the ant colony algorithm [47, 48], the particle swarm algorithm [49–52], and the artificial bee colony algorithm [53], have been studied recently in ASP.

SFLA is an evolution algorithm that is used to calculate the global optima of several combinatorial problems and has been found to be effective in searching for global solutions. Rahimi-Vahed and Mirzaei proposed a hybrid multiobjective SFLA for a mixed model assembly line sequencing problem [54] and bicriteria permutation flow shop scheduling problem [55]. Li et al. [56] proposed an effective SFLA for a multiobjective flexible job shop scheduling problems. Fang and Wang [57] introduced an effective SFLA for a resource-constrained project scheduling problem. Li et al. [58] developed a modified SFLA for continuous optimization. The aforementioned studies show that SFLA is a simple, robust, and fast algorithm for solving combinatorial optimization problems.

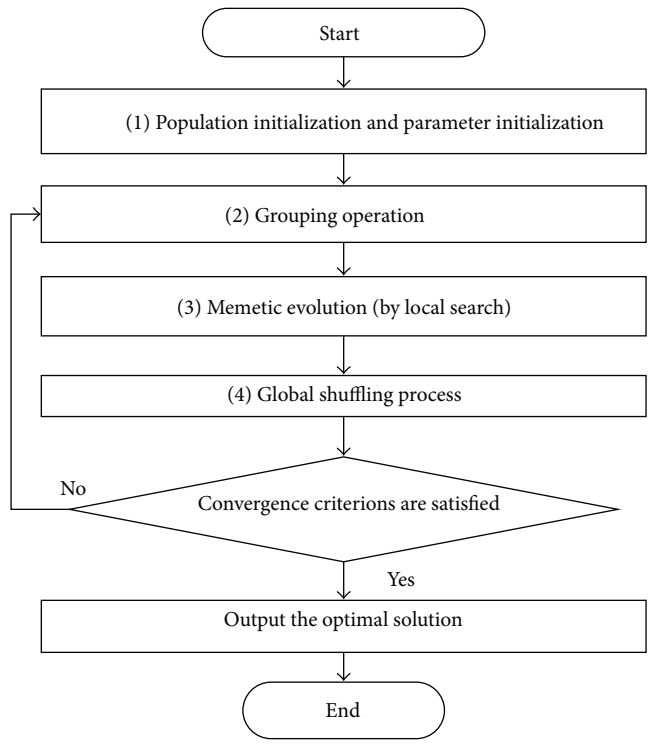


Figure 1: SFLA procedure.

3. SFLA

SFLA is a metaheuristic optimization method that identifies solutions by simulating collaboration behaviors and interactive information similar to a frog community searching for food in a natural environment. This algorithm divides population into several subpopulations, and the evolution of memes is driven by the exchange of global information among subpopulations and the local evolutionary search within subpopulation.

SFLA is described in detail as follows. The population consists of several frogs, and each frog is a solution to the problem. The population is divided into several subpopulations through a grouping operator to simulate frog grouping behaviors. Each subpopulation is called a memeplex. A memeplex is composed of frogs with the same meme that perform local searches. Each frog has its own idea but is also influenced by other frogs in the same memeplex. Frogs adjust their positions through memetic evolution. After a predefined number of memetic evolutionary times, frogs in different subpopulations exchange information through a global shuffling process. The alternating memetic evolution and global shuffling process make the frogs jump out of the local optimum and evolve toward the global optimum.

The SFLA procedure is illustrated in Figure 1. The detailed procedure is as follows.

(1) Population initialization and parameter initialization: SFLA initially creates a population of F frogs as a certain solution amount. The ith frog (i.e., the ith solution to the problem) is represented as F_i =

$(f_{i1}, f_{i2}, \ldots, f_{is})$, where s is the solution space dimension.

(2) Grouping operator: a grouping operator separates F frogs into m memeplex, which contains n frogs, according to their fitness order. The frogs are sorted in descending order according to their fitness. The first frog is assigned to the first memeplex, the second frog is assigned to the second memeplex, the mth frog is assigned to the mth memeplex, the $(m+1)$th frog is assigned back to the first memeplex, and so on. The best and worst positions of each frog in each memeplex are indicated as F_b and F_w, respectively, and the frog in the best position in the entire population is indicated as F_g.

(3) Memetic evolution: during the memetic evolution of the memeplex, the frog in the worst position F_w is updated through a local search in its memeplex. The new position of the worst frog is updated according to (1) to (3):

$$D = M\left(\text{rand}(0,1) * \left(F_b - F_w\right)\right), \quad (1)$$

$$F_{w,\text{new}} = F_{w,\text{old}} + D\left(-D_{\max} \le D \le D_{\max}\right), \quad (2)$$

$$D = M\left(\text{rand}(0,1) * \left(F_g - F_w\right)\right), \quad (3)$$

where D is the moving distance matrix, $M(\text{rand}(0,1))$ is the random number matrix within the range $[0,1]$, and $D_{\max}$ is the maximum distance that the frog is permitted to move.

The memetic evolution procedure in SFLA is illustrated in Figure 2.

If the new position of the worst frog is better than its previous position after (1) and (2) are calculated, then it replaces the position of the worst frog and the worst position F_w is recalculated. Otherwise, substitute F_g for F_b and repeat the worst position of the frog updating calculations in (3) and (2). If the frog in the worst position still cannot obtain a better position, then a frog with a new position is stochastically produced to replace the frog in the worst position F_w. Consequently, each memeplex follows a predefined memetic evolutionary time.

(4) Global shuffling process: all frogs are mixed and sorted in descending order according to their fitness. The memeplex is then divided according to the new order, and then Step (2) is repeated.

(5) Local evolution and global shuffling processes continue until convergence is achieved.

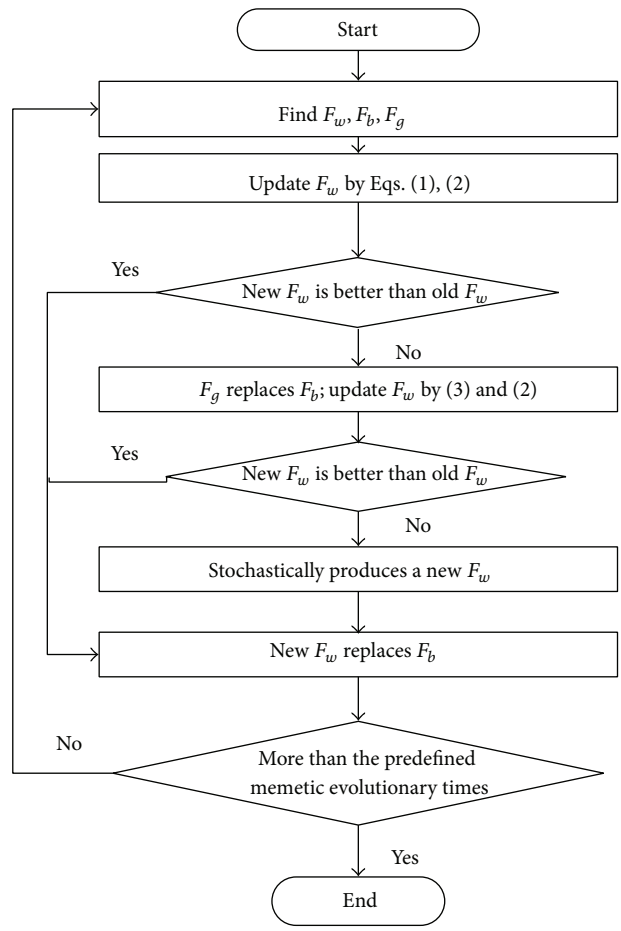


Figure 2: Memetic evolution procedure in SFLA.

4. ASP Problem Statement

The final goal of ASP is to enhance assembly efficiency and reduce assembly difficulties and costs. Several parameters are involved to achieve this objective. This study employs several essential parameters as evaluating indicators, including geometric feasibility, assembly stability, changing times of assembly tool, and changing times of assembly direction. The fitness function is ultimately developed through the evaluating indicators.

4.1. Geometric Feasibility. The assembly direction is divided into six types of direction, as follows: $d(k) = \{+x, +y, +z, -x, -y, -z\}$. Interference values I_{ijd_k} ($k = 1, 2, \ldots, 6$ and $d_k \in d(k)$) describe whether part P_i interfere with part P_j when moving along d_k direction. I_{ijd_k} is as follows:

$$I_{ijd_k} = \begin{cases} 0, & p_i \text{ does not interfere with } p_j \text{ in the } d_k \text{ direction} \\ 1, & p_i \text{ interferes with } p_j \text{ in the } d_k \text{ direction.} \end{cases} \quad (4)$$

Suppose that $\text{AP} = (P_1, P_2, \ldots, P_n)$ is an assembly sequence. The part set $\text{AP}_1 = \{P_1, P_2, \ldots, P_{i-1}\}$ is the set in which parts have been assembled, and P_i is the part to be assembled. Then $S_k(P_i)$ ($k = 1, 2, \ldots, 6$) is the sum of the

```
IF c_{P_i P_j} = 2, j ∈ [1, i − 1] exists, THEN
    Assembly operation is stable;
ELSE
IF c_{P_i P_j} == 0, j ∈ [1, i − 1], THEN
    Assembly operation is unstable;
ELSE
    IF s_{P_j P_i} = 1, j ∈ [1, i − 1] exists, THEN
        Assembly operation is stable;
    ELSE IF s_{P_j P_i} == 0, THEN
        Assembly operation is unstable
    ENDIF
ENDIF
```

Algorithm 1: Stability evaluation method of the assembly sequence.

interference values which is P_i and each part in AP_1 when P_i is assembled along d_k direction. $S_k(P_i)$ is as follows:

$$S_k(P_i) = \sum_{j=1}^{i-1} I_{P_i P_j d_k}. \tag{5}$$

If $S_k(P_i) = 0$, then P_i can be assembled along d_k. Otherwise, P_i cannot be assembled along d_k. In this case, we obtain $DC(P_i) = \{d_k \mid S_k(P_i) = 0\}$ that is the feasible assembly direction set of P_i. For each P_i $(1 < i \le n)$, if $DC(P_i) \ne \phi$, then AP is the feasible assembly sequence; otherwise, AP is the infeasible assembly sequence. n_f is expressed as the total times of assembly interference of AP. The value of n_f is equal to the total times of $DC(P_i) = \phi$ $(1 < i \le n)$ in AP.

4.2. Assembly Stabilities. In the actual assembly process, parts may become unstable because of gravity. Several assembly operations must use a jig or auxiliary tool to maintain stability when a part is unstable during the assembly process, which results in an inefficient assembly. Therefore, the stability of the assembly sequence should be evaluated.

The augmented adjacency matrix $C = (c_{ij})_{n\times n}$ and support matrix $S = (s_{ij})_{n\times n}$ are defined to evaluate the stability of the assembly sequence. In the augmented adjacency matrix, c_{ij} expresses the connection type between P_i and P_j. For a stable connection, $c_{ij} = 2$; for a contact connection, $c_{ij} = 1$; and for a noncontact connection, $c_{ij} = 0$. In the support matrix, s_{ij} expresses the support type between P_i and P_j. For stable support, $s_{ij} = 1$; otherwise, $s_{ij} = 0$.

Suppose that $\mathrm{AP} = (p_1, p_2, \ldots, p_n)$ is an assembly sequence. The part set $\mathrm{AP}_1 = \{p_1, p_2, \ldots, p_{i-1}\}$ is expressed as the parts having been assembled, and P_i is expressed as the part to be assembled. The stability evaluation method of the assembly sequence is shown in Algorithm 1. In this study, n_s expresses the times of the assembly sequence stable operation. A smaller n_s indicates a more stable assembly sequence.

4.3. Changing Times of Assembly Tool. Given the particularity of each assembly part, different assembly tools should be used in the actual assembly process. Changing the assembly tool leads to a long assembly time and high cost for the assembly process. Therefore, changing times of assembly tool should be as few as possible.

Suppose the assembly sequence is $\mathrm{AP} = (P_1, P_2, \ldots, P_n)$ and assembly tool sequence of AP is Tc. $Tc(P_i)$ is expressed as the assembly tool of P_i. Assembly tool of each part is determined by the characteristic of each part and the available assembly tool. The assembly tool sequence for an AP, as well as the optimal assembly tool sequence, is predetermined. Changing times of assembly tool n_t are calculated as shown in Algorithm 2.

4.4. Changing Times of Assembly Direction. The reduced changing times of assembly direction shorten assembly time and enhance assembly efficiency. Supposing the assembly sequence is $\mathrm{AP} = (P_1, P_2, \ldots, P_n)$, changing times of assembly direction n_d are calculated as shown in Algorithm 3.

4.5. Fitness Function. Different radioactive equipment under various environments may have varying influence degrees for the evaluating indicators. Therefore, weighting factors must be determined according to the actual situation. A penalty function $c_f n_f$ is applied to infeasible assembly sequence to speed up the algorithm convergence rate. Then the weighted fitness function is as follows:

$$f = c_s n_s + c_t n_t + c_d n_d + c_f n_f, \tag{6}$$

where c_s, c_t, c_d, and c_f are the weighting factors for each evaluating indicator, and c_f must be generally larger than the other three weighting factors (i.e., $c_f \ge (n/2)\max\{c_s, c_t, c_d\}$). In this study, a small fitness function value indicates good position of the frog and good assembly sequence.

5. Improved SFLA for ASP

5.1. Local Search Strategy Based on a Swap Sequence. ASP is a combinatorial optimization problem in which each solution dimension is discrete. A GA can solve the discrete optimization problem by using crossover and mutation operators. The improved SFLA introduces a local search strategy based on a swap sequence to address this problem.

5.1.1. Swap Factor and Swap Sequence

(1) Swap Factor. Suppose that an assembly sequence that includes n parts is expressed as $\mathrm{AP} = (P_1, P_2, \ldots, P_n)$. The function of swap factor $vo(\gamma, \omega)$ is to swap the positions of γ and ω to form a new assembly sequence. For example, if the initial assembly sequence is $\mathrm{AP} = (2, 4, 3, 5, 1)$ and the swap factor is $vo = vo(2, 4)$, then $\mathrm{AP}' = \mathrm{AP} \oplus vo = (2, 5, 3, 4, 1)$. $\oplus$ indicates that the swap factor is acting on the assembly sequence.

(2) Swap Sequence. $vos = (vo_1, vo_2, \ldots, vo_n)$ expresses a swap sequence that consists of n swap factors, in which $vo_1, vo_2, \ldots, vo_n$ are the swap factors and their order does not satisfy the commutative law. The effect of a swap sequence on

Step 1. **Set** $i = 0, m = 1, n_t = 0$
Step 2. **IF** $\bigcap_i^{i+m} Tc(P_i) = \varnothing$ **THEN**
/* Assembly tool must be changed when P_i is to be assembled. */
$n_t = n_t + 1$
Proceed to Step 4
ELSE
Proceed to Step 3
ENDIF
Step 3. **Set** $m = m + 1$
IF $i + m \leq n$, **THEN**
Proceed to Step 2
ELSE
Proceed to Step 5
ENDIF
Step 4. **IF** $i + m = n$, **Then**
Proceed to Step 5
ELSE
Set $i = i + m, m = 1.$
Proceed to Step 2
ENDIF
Step 5. **Output** n_t

Algorithm 2: Calculation procedure for changing times of assembly tool.

Step 1. **Set** $i = 0, m = 1, n_d = 0$
Step 2. **IF** $\bigcap_i^{i+m} Dc(P_i) = \varnothing$ **THEN**
/* Assembly direction must be changed when P_i is to be assembled. */
$n_d = n_d + 1$
Proceed to Step 4
ELSE
Proceed to Step 3
ENDIF
Step 3. **Set** $m = m + 1$
IF $i + m \leq n$, **THEN**
Proceed to Step 2
ELSE
Proceed to Step 5
ENDIF
Step 4. **IF** $i + m = n$, **Then**
Proceed to Step 5
ELSE
Set $i = i + m, m = 1.$
Proceed to Step 2
ENDIF
Step 5. **Output** n_d

Algorithm 3: Calculation procedure for changing times of assembly direction.

an assembly sequence is equal to the effect of each swap factor in a swap sequence on the assembly sequence.

AP_a and AP_b are two assembly sequences. $vos(\mathrm{AP}_b \Theta \mathrm{AP}_a)$ expresses the swap sequence in which AP_a is adjusted as AP_b. It can be expressed as (5):

$$\mathrm{AP}_b = \mathrm{AP}_a \oplus vos\left(\mathrm{AP}_b \Theta \mathrm{AP}_a\right) = \mathrm{AP}_a \oplus \left(vo_1, vo_2, \ldots, vo_n\right). \tag{7}$$

A false code of the swap sequence is shown in Algorithm 4. For example, if $\mathrm{AP}_a = (1, 4, 2, 5, 3)$ and $\mathrm{AP}_b = (2, 3, 5, 1, 4)$, then the swap sequence is $vos(\mathrm{AP}_b \Theta \mathrm{AP}_a) = (vo_1(1, 3), vo_2(2, 5), vo_3(3, 4))$.

5.1.2. Frog Position Updating Strategy. $\|D\|$ is the number of swap factors contained by the moving distance matrix D. ceil is the top integral function. rD ($r \in [0, 1]$) is the first ceil($r\|D\|$) swap factors in the swap sequence D. For example, if $D = vos = (vo_1, vo_2, vo_3, vo_4)$ and $r = 0.6$, then $\|D\| = 4$ and $rD = (vo_1, vo_2, vo_3)$.

Step 1. **Set** $i = 1$, $j = 0$
Step 2. **IF** $\mathrm{AP}_a(i) \neq \mathrm{AP}_b(i)$, **THEN**
$j = j + 1$
$v = find(\mathrm{AP}_b == \mathrm{AP}_a(i))$
$vo_j = vo(i, v)$
$\mathrm{AP}_a = \mathrm{AP}_a \oplus vo_j$
ENDIF
Step 3. **Set** $i = i + 1$,
IF $i \leq n - 1$, **THEN**
Proceed to Step 2
ELSE
Proceed to Step 4
ENDIF
Step 4. $vos = (vo_1, vo_2, \ldots, vo_j)$

Algorithm 4: False code of the swap sequence.

Equations (1) to (3) are improved as follows:

$$D = M\left(\mathrm{rand}\,(0,1) * \left(F_b \Theta F_w\right)\right), \tag{8}$$

$$F_{w,\mathrm{new}} = F_{w,\mathrm{old}} \oplus D\left(\left\|D_{\min}\right\| \leq \|D\| \leq \left\|D_{\max}\right\|\right), \tag{9}$$

$$D = M\left(\mathrm{rand}\,(0,1) * \left(F_g \Theta F_w\right)\right). \tag{10}$$

5.2. Diversity Control Strategy. After population sorting in SFLA, the grouping operator makes the best frog position similar in each memeplex when the first m frogs satisfy $F_1 = F_2 = \cdots = F_m$ (m is the quantity of the memeplex). Based on (8) to (10), each memeplex can easily converge to the best frog position F_g of the entire population. The algorithm search space and the probability of algorithm convergence with the globally optimal solution are reduced. This study proposes a diversity control strategy to avoid homoplasy. The control policy is as follows.

(1) Compute N of the preceding same frog position after the grouping operator.

(2) If $N < m$, then proceed to step (4).

(3) The next population is based on standard GA.

(4) The population is based on the other SFLA steps.

5.3. Improved SFLA Steps. The basic steps to solve the ASP problem by using the improved SFLA are shown in Figure 3. The detailed steps are as follows.

(1) Parameter initialization and population initialization: frog population size is *size*. The number of frog memeplex is m. The population iterative is *iter*. The local search iteration is *mrun*. The maximum and minimum frog moving distances are $\|D_{\max}\|$ and $\|D_{\min}\|$, respectively. The crossover probability is *pc*. The adaptive mutation probability is *pm*.

(2) Modified grouping operator: the frogs are sorted in descending order according to their fitness during the preprocessing of the grouping operator in SFLA. In ASP, a small fitness function value indicates good frog position and good assembly sequence. Therefore, the grouping operator should be modified. The modified grouping operator is as follows. Suppose that the scale of the frog population is N, which is then divided into m memeplexes. All frogs in the population are arranged in ascending order according to the fitness function value. The first frog enters the first memeplex and the second frog enters the second memeplex, and so on, until the m frog enters the mth memeplex. The $(m + 1)$th frog is then assigned back to the first memeplex, and so on. All individual frogs are assigned according to the aforementioned rule.

(3) Memetic evolution (local search based on a swap sequence): in the improved SFLA, memetic evolution is modified and performed by using a local search strategy based on a swap sequence until the *mrun* generation.

(4) Optimal sampling different strategy: the optimal sampling different strategy is included in the improved SFLA to avoid homoplasy for each memeplex.

(5) Global shuffling process: the global shuffling process of the improved SFLA is similar to SFLA and updates the best position F_g of the frog population.

(6) The next step is determining whether the iteration should be terminated according to the terminal condition of the algorithm. If the terminal condition is

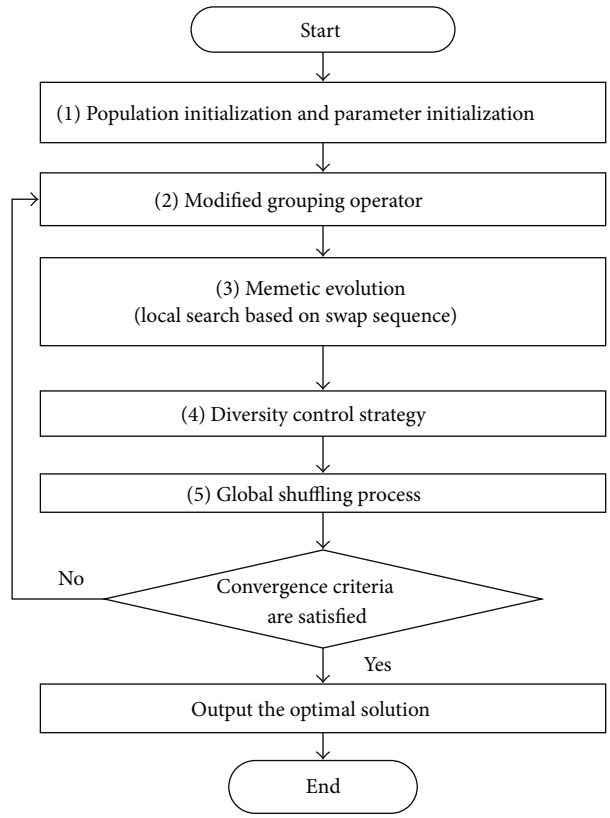


Figure 3: Memetic evolution procedure in SFLA.

Table 1: Part assembly tool sets.

Part number	Part name	Assembly tool
1	Nut washer assembly 1	$T1$
2	Nut washer assembly 2	$T1$
3	Nut washer assembly 3	$T1$
4	Nut washer assembly 4	$T1$
5	Hydraulic cylinder	$T4$
6	Pole 1	$T3$
7	Pole 2	$T3$
8	Pole 3	$T3$
9	Pole 4	$T3$
10	Strut 1	$T4$
11	Nut 1	$T1$
12	Nut 2	$T1$
13	Nut 3	$T1/T2$
14	Bolt 1	$T1/T3$
15	Bolt 2	$T1/T3$
16	Pin 1	$T3$
17	Nut 4	$T1/T2$
18	Pin 2	$T3$
19	Central pin	$T3$
20	Back plate	$T4$
21	Strut 2	$T4$
22	Nut washer assembly 5	$T1$
23	Nut washer assembly 6	$T1$
24	Nut washer assembly 7	$T1$
25	Nut washer assembly 8	$T1$
26	Axis	$T3$
27	Hydraulic pressure scissors	$T4$
28	Hydraulic pressure shear blades 1	$T5$
29	Hydraulic pressure shear blades 2	$T5$

satisfied, then the iteration ends. Otherwise, return to step (2).

6. Experiment and Analysis

The application program based on the improved SFLA is compiled under MATLAB environment. The computer environment of the application program consists of a 2.0 GHz CPU, 2 GB memory, and Windows 7 32-bit operating system. The hydraulic pressure shear, which contains 29 parts, is used for the ASP experiment. The exploded view of the hydraulic pressure shear is shown in Figure 4. The components of the assembly tool sets are listed in Table 1.

6.1. ASP Experiment Based on SFLA. After conducting an orthogonal experiment on the assembly of the hydraulic pressure shear, the algorithm rapidly identifies an optimal assembly sequence when the weighting factors of the evaluating indicator in the fitness function are $c_f = 4$, $c_s = 0.5$, $c_t = 0.2$, and $c_d = 0.3$. If the memeplex has few local search iterations, then it also undergoes few evolution times, which reduces information exchange within the memeplex. If the memeplex has many local search iterations, then it undergoes multiple local searches, which increases algorithm search time and makes the best frog position of various memeplexes similar. It also causes the algorithm to carry out GA several times and thus slows down the convergence rate of the algorithm. If the maximum moving distance $D_{\max}$ is too small, then the global algorithm search capability is reduced. It causes the algorithm to easily fall into a local search. If $D_{\max}$ is too large, then the algorithm is unable to find the globally optimal solution. After multiple comparison experiments, the algorithm optimization capability is observed to be optimal when the local search iteration of the memeplex is 10, crossover probability is 0.8, adaptive mutation probability is 0.1, maximum moving distance is 8, minimum moving distance is 1, and frog quantity in the memeplex is maintained at 30.

The ASP experiment is conducted with population sizes 60, 120, 180, and 240 given that each parameter value of the improved SFLA and the weighting factor of the evaluating indicator in the fitness function are the same. The fitness function value distributed the optimal assembly sequence from the results of the ASP experiment. The analysis results are shown in Figure 5, in which the number of algorithm iterations is 600 and that of repeating operation times is 50. A lot of experiments show that the fitness value of the global optimal assembly sequence F is 2.1. As shown in Figure 5, the distributed situation of the local optimal fitness value is within the following ranges: 2.1 to 3.0, 3.1 to 4.0, 4.1 to 5.0, 5.1 to 6.0, and >6.0. As shown in Figure 5, the distributed situations of fitness value of local optimal assembly sequence differ along with various population sizes. When population size is 60 and time of experiment is 50, there is only one fitness value of local optimal assembly sequence in sections 2.1 to 3.0. As population size increases, the quantity of fitness value of local optimal assembly sequence identified by the algorithm in this section gradually increases. When population size increases to 240, the quantity identified by the algorithm in this section is 20. As population size increase, the quantity of outstanding assembly sequences whose fitness value is smaller gradually increases. As shown in Table 2, the increase in algorithm population size reduces algorithm iteration efficiency, and the operation time of the algorithm is extended. In this experiment, the probability of the global optimal assembly sequence identified by the algorithm is highest when population size is 240 and the average consuming time of a single experiment is in the range of acceptable with 515S.

The mean and optimal average fitness of iteration in 50 experiments when population size is 120 are shown in Figure 6. As algorithm iteration increases, the optimal fitness average value fluctuates. However, the optimal and average values of the average fitness steadily decrease from the overall tendency, which shows that the stability of the algorithm is good and the algorithm convergence rate at the later period is slow.

The mean and optimal average fitness of 1 of the 50 experiments in which the global optimal assembly sequence is obtained when population size is 120 is shown in Figure 7. In the algorithm implementation, the mean fitness exhibits a

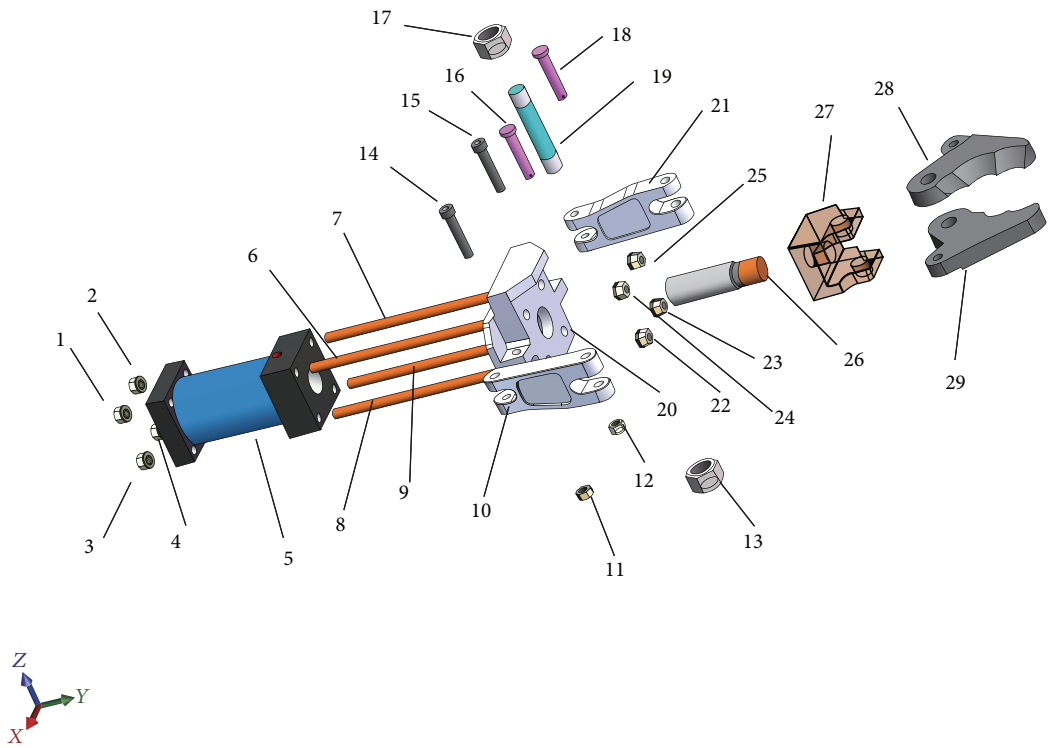


Figure 4: Exploded view of the hydraulic pressure shear.

huge fluctuation because the optimal sample is homoplasy. According to the diversity control strategy, the algorithm executes GA and increases average fitness. The homoplasy of the algorithm improves after executing GA.

6.2. Algorithm Comparison Experiment. An algorithm comparison experiment is conducted among improved SFLA (SFLA-GA), GA, SFLA, PSO, and AMPSO [7] to verify the performance of the improved SFLA for the ASP problem. AMPSO is modified method for ASP in RHM in our previous work [7].

A hydraulic pressure shear is employed to carry out ASP with the same programming and PC environments as those indicated in Section 6.1. Moreover, the parameter $f_d = 2.1$ and values of other related parameters in AMPSO are the same as those presented in [11]. Inertia weight in PSO is 0.6, and the values of the other parameters are similar to those of AMPSO. The experiment results with population sizes of 60 and 240 are shown in Table 3. The algorithm convergence curves of different population sizes are shown in Figures 8 and 9 (i.e., variation of population average fitness values along with the iterations).

As shown in Table 3, the probability of a feasible assembly sequence identified by the improved SFLA is enhanced when population size increases. The probability is higher than those for SFLA and GA but lower than those for AMPSO and PSO. Under the same population size, the improved SFLA exhibits a superior assembly sequence than those of GA and SFLA. The value of the optimal assembly sequence fitness function identified by the improved SFLA is less than those identified by GA and SFLA. When population size is 60, the improved SFLA obtains an acceptable assembly sequence and the fitness function value is 2.9. However, GA and SFLA are unable to obtain an acceptable assembly sequence even when the population size is 240. The fitness values of the optimal assembly sequence are similar in PSO and AMPSO. The execution time of the improved SFLA is slightly less than that of GA and significantly less than those of AMPSO and PSO. Consequently, the efficiency of the proposed algorithm is acceptable. Based on the local optimal fitness average value of the algorithm, the improved SFLA exhibits a higher convergence rate than those of GA and SFLA and is near those of AMPSO and PSO. The improved SFLA is even better than PSO when population size is 240. Therefore, the optimization capability, efficiency, and convergence rate of the improved SFLA are better than those of GA, whereas its optimization capability and convergence rate are better than those of SFLA. The overall performance of the SFLA-GA for solving ASP problems proposed in this study is similar to those of AMPSO and PSO. As shown in Figures 8 and 9, the stochastic initializing population qualities of the five algorithms are approximately similar. Therefore, the preceding analysis is reliable.

7. Conclusions

RHM is an important mean of ensuring the reliability of radioactive equipment and has a wide application in radioactive installations. RHMP predetermine the maintenance procedures of radioactive equipment during the design

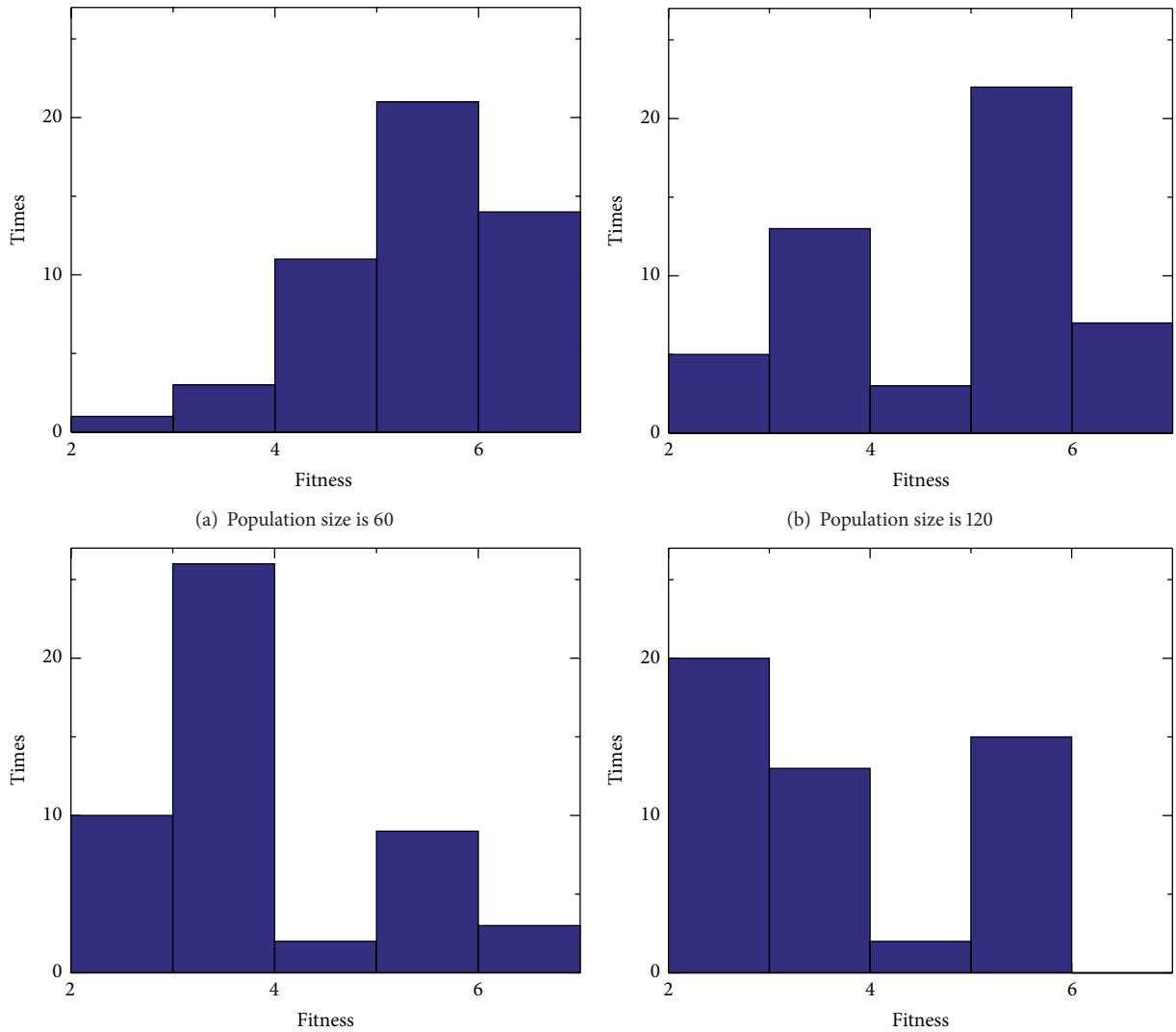
Table 2: Execution of the 50 optimal ASP comparison results under different population sizes.

Population size	60			120			180			240		
	S	D	T	S	D	T	S	D	T	S	D	T
Assembly sequence related information	20	$-X$	$T4$	5	$-Y$	$T4$	20	$-Y$	$T4$	20	$-X$	$T4$
	21	$-X$	$T4$	20	$-Y$	$T4$	10	$-Y$	$T4$	10	$-X$	$T4$
	10	$-X$	$T4$	10	$-Y$	$T4$	21	$-Y$	$T4$	5	$-X$	$T4$
	5	$-X$	$T4$	21	$-Y$	$T4$	26	$-Y$	$T3$	21	$-Y$	$T4$
	26	$-Y$	$T3$	7	$-Y$	$T3$	27	$-Y$	$T4$	8	$-Y$	$T3$
	27	$-Y$	$T4$	6	$-Y$	$T3$	29	$-Y$	$T5$	26	$-Y$	$T3$
	29	$-Y$	$T5$	9	$-Y$	$T3$	28	$-Y$	$T5$	6	$-Y$	$T3$
	28	$-Y$	$T5$	26	$-Y$	$T3$	5	$-Z$	$T4$	9	$-Y$	$T3$
	23	$-Z$	$T1$	8	$-Y$	$T3$	19	$-Z$	$T3$	7	$-Y$	$T3$
	15	$-Z$	$T1$	22	$-Y$	$T1$	16	$-Z$	$T3$	22	$-Y$	$T1$
	25	$-Z$	$T1$	24	$-Y$	$T1$	18	$-Z$	$T3$	25	$-Y$	$T1$
	22	$-Z$	$T1$	23	$-Y$	$T1$	23	$-Z$	$T1$	24	$-Y$	$T1$
	24	$-Z$	$T1$	25	$-Y$	$T1$	15	$-Z$	$T1$	23	$-Y$	$T1$
	19	$-Z$	$T3$	27	$-Y$	$T4$	25	$-Z$	$T1$	27	$-Y$	$T4$
	9	$+Y$	$T3$	28	$-Y$	$T5$	22	$-Z$	$T1$	28	$-Y$	$T5$
	6	$+Y$	$T3$	29	$-Y$	$T5$	17	$-Z$	$T1$	29	$-Y$	$T5$
	8	$+Y$	$T3$	16	$-Z$	$T3$	14	$-Z$	$T1$	14	$-Z$	$T3$
	7	$+Y$	$T3$	18	$-Z$	$T3$	24	$-Z$	$T1$	18	$-Z$	$T3$
	1	$+Y$	$T1$	14	$-Z$	$T3$	6	$+Y$	$T3$	19	$-Z$	$T3$
	4	$+Y$	$T1$	19	$-Z$	$T3$	9	$+Y$	$T3$	16	$-Z$	$T3$
	3	$+Y$	$T1$	17	$-Z$	$T1$	7	$+Y$	$T3$	17	$-Z$	$T1$
	2	$+Y$	$T1$	15	$-Z$	$T1$	8	$+Y$	$T3$	15	$-Z$	$T1$
	12	$+Z$	$T1$	12	$+Z$	$T1$	1	$+Y$	$T1$	11	$+Z$	$T1$
	13	$+Z$	$T1$	11	$+Z$	$T1$	3	$+Y$	$T1$	13	$+Z$	$T1$
	11	$+Z$	$T1$	13	$+Z$	$T1$	2	$+Y$	$T1$	12	$+Z$	$T1$
	17	$-Z$	$T1$	3	$+Y$	$T1$	4	$+Y$	$T1$	2	$+Y$	$T1$
	16	$-Z$	$T3$	1	$+Y$	$T1$	12	$+Z$	$T1$	4	$+Y$	$T1$
	14	$-Z$	$T3$	4	$+Y$	$T1$	11	$+Z$	$T1$	3	$+Y$	$T1$
	18	$-Z$	$T3$	2	$+Y$	$T1$	13	$+Z$	$T1$	1	$+Y$	$T1$
Changing times of assemble direction	5			3			3			4		
Changing times of assemble tool	7			6			8			6		
Unstable operation times	0			2			0			0		
Single execution time/s	135			256			387			515		
Fitness	2.9			2.1			2.5			2.4		

Note: S represents the assembly sequence, D represents the assembly direction, and T represents the assembly tool.

of radioactive installation. As a part of RHMP, ASP is introduced in this study. Evolution algorithm is a useful tool for ASP which is considered as a combinatorial optimization problem. The contribution of this study is to develop an advanced evolution algorithm named improved SFLA for ASP.

SFLA is an evolution algorithm that is used to calculate the global optima of several combinatorial problems and has been found to be effective in searching for global solutions. There are mainly two improvement strategies that were employed in our improved algorithm: (1) each SFLA operation was redefined with respect to the discreteness characteristic of ASP; (2) a diversity control strategy based on GA was introduced to avoid homoplasy for each memeplex. The experiments proved that the global optimization capability and convergence rate of the improved SFLA are better than those of SFLA and GA and similar to those of AMPSO and PSO. Moreover, the algorithm operation



(a) Population size is 60

(b) Population size is 120

(c) Population size is 180

(d) Population size is 240

Figure 5: Distributed situations of each local optimal fitness.

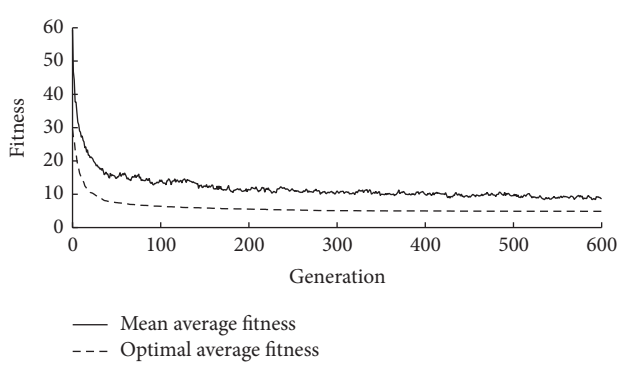


Figure 6: Mean and optimal average fitness of iteration in 50 experiments.

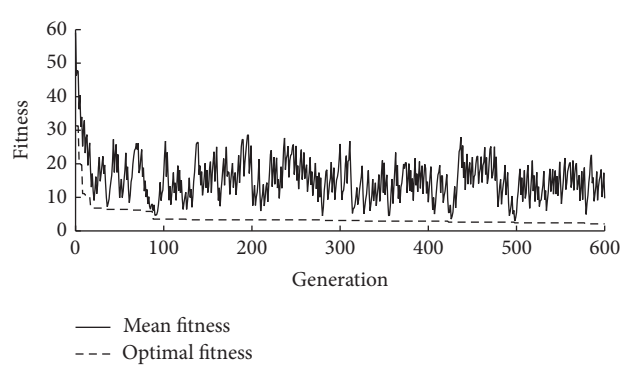


Figure 7: Mean and optimal fitness.

efficiency should be better than GA to achieve an enhanced assembly sequence result. Experiment results showed that the proposed algorithm is an advanced evolution algorithm and exhibits outstanding performance in solving the ASP

Table 3: Comparison test results of the algorithm.

Parameter	GA		SFLA		AMPSO		PSO		SFLA–GA	
Population size	60	240	60	240	60	240	60	240	60	240
Iteration times	600	600	600	600	600	600	600	600	600	600
Execution times	50	50	50	50	50	50	50	50	50	50
Feasible sequence number	3	1	0	1	33	40	29	40	19	37
Running time/s	7456	29586	6920	27762	8578	34822	8484	34178	6761	25763
Optimal assembly sequence fitness value	7.0	6.4	6.8	6.5	2.4	2.4	3.50	2.3	2.9	2.4
Local optimal-fitness average value	10.08	8.17	12.77	9.98	5.27	3.79	5.16	4.13	5.77	3.74

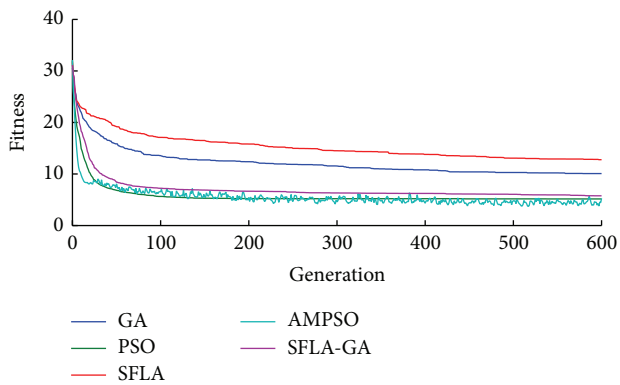


Figure 8: Convergence curve (population size = 60).

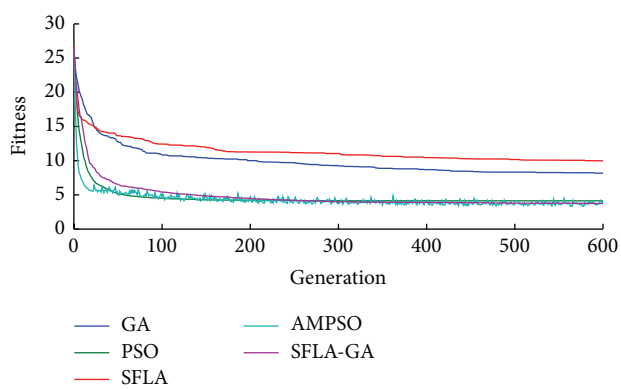


Figure 9: Convergence curve (population size = 240).

problem. The application of the proposed algorithm should increase the level of ASP in a radioactive environment. However, the experiments also proved that the convergence rate of the improved SFLA at the later period is slow. Further works should be conducted to improve the proposed algorithm and enhance its convergence rate.

Notations

AMPSO: Adaptive mutation particle swarm optimization
AP: Assemble sequence
ASP: Assembly sequence planning
c_{ij}: Connection type between part P_i and part P_j
$DC(P_i)$: Feasible assembly direction set of part P_i in an assemble sequence
d_k: The kth direction in the set $d(k) = \{+x, +y, +z, -x, -y, -z\}$
$D_{\max}$: Maximum distance that the frog is permitted to move
F_b: The best positions of each frog in each memeplex
F_w: The worst positions of each frog in each memeplex
GA: Genetic algorithm
I_{ijd_k}: Whether part P_i interfere with part P_j when moving along d_k direction
ITER: International thermonuclear experimental reactor
n_d: Changing times of assembly direction
n_f: Total times of assembly interference in an assembly sequence
n_s: Times of the assembly sequence stable operation
n_t: Changing times of assembly tool
P_i: The ith part in an assemble sequence
PSO: Particle swarm optimization
RHM: Remote handling maintenance
RHMP: Remote handling maintenance planning
SFLA: Shuffled frog leaping algorithm
SFLA-GA: Improved shuffled frog leaping algorithm proposed by this paper
s_{ij}: Support type between part P_i and part P_j
$S_k(P_i)$: Sum of the interference values
$Tc(P_i)$: The assembly tool of part P_i
$vo(\gamma, \omega)$: Swap factor to swap the positions of γ and ω to form a new assembly sequence
vos: Swap sequence which is swap factors set.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The study was supported by the National Natural Science Foundation of China (Grant no. 71201026), the Project of the Department of Education of Guangdong Province (no. 2013KJCX0179, no. 2014KTSCX184, and no. 2014KGJHZ014), the Development Program for Excellent Young Teachers in Higher Education Institutions of Guangdong Province (no. Yq2013156), the China Spallation Neutron Source Electromechanical Technology R&D Joint Laboratory Foundation (no. ZD120512), the Dongguan Social Science and Technology Development Project (no. 2013108101011), and the Dongguan Universities and Scientific Research Institutions Science and Technology Project (no. 2014106101007).

References

[1] J. L. Seminara and S. O. Parsons, "Nuclear power plant maintainability," *Applied Ergonomics*, vol. 13, no. 3, pp. 177–189, 1982.

[2] W. B. Scott, W. I. Enderlin, A. D. Chockie, and K. A. Bjorkelo, "Good practices for effective maintenance to manage aging of nuclear power plants," *Nuclear Engineering and Design*, vol. 134, no. 2-3, pp. 257–265, 1992.

[3] S. Martorell, V. Serradell, and G. Verdú, "Safety-related equipment prioritization for reliability centered maintenance purposes based on a plant specific level 1 PSA," *Reliability Engineering and System Safety*, vol. 52, no. 1, pp. 35–44, 1996.

[4] H. Kinoshita, M. Teshigawara, M. Ito et al., "Remote handling devices in MLF," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 600, no. 1, pp. 78–80, 2009.

[5] EUROfusion, https://www.euro-fusion.org/fusion/jet-remote-handling/what-is-remote-handling/.

[6] I. Ribeiro, C. Damiani, A. Tesini, S. Kakudate, M. Siuko, and C. Neri, "The remote handling systems for ITER," *Fusion Engineering and Design*, vol. 86, no. 9-10, pp. 471–477, 2011.

[7] J.-W. Guo, S. Wang, H.-B. Chen, Z.-Z. Sun, and Z.-C. Zhang, "An intelligent approach for remote handling maintenance sequence planning in radiation environment," *Journal of Chemical and Pharmaceutical Research*, vol. 6, no. 4, pp. 543–552, 2014.

[8] L. Wang, S. Keshavarzmanesh, H.-Y. Feng, and R.-O. Buchal, "Assembly process planning and its future in collaborative manufacturing: a review," *The International Journal of Advanced Manufacturing Technology*, vol. 41, no. 1-2, pp. 132–144, 2009.

[9] R. H. Wilson and J.-C. Latombe, "Geometric reasoning about mechanical assembly," *Artificial Intelligence*, vol. 71, no. 2, pp. 371–396, 1994.

[10] L. Kavraki, J.-C. Latombe, and R. H. Wilson, "On the complexity of assembly partitioning," *Information Processing Letters*, vol. 48, no. 5, pp. 229–235, 1993.

[11] P. Jiménez, "Survey on assembly sequencing: a combinatorial and geometrical perspective," *Journal of Intelligent Manufacturing*, vol. 24, no. 2, pp. 235–250, 2013.

[12] M. M. Eusuff and K. E. Lansey, "Optimization of water distribution network design using the shuffled frog leaping algorithm," *Journal of Water Resources Planning and Management*, vol. 129, no. 3, pp. 210–225, 2003.

[13] M. Eusuff, K. Lansey, and F. Pasha, "Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization," *Engineering Optimization*, vol. 38, no. 2, pp. 129–154, 2006.

[14] Japan Atomic Energy Agency, *Technical Design Report of Spallation Neutron Source Facility in J-PARC*, Japan Atomic Energy Agency, Ibaraki-ken, Japan, 2012.

[15] Z. Zhou, D. Yao, and P. Zi, "The research activities on remote handling system for CFETR," *Journal of Fusion Energy*, vol. 34, no. 2, pp. 232–237, 2015.

[16] K. Kershaw, B. Feral, J.-L. Grenard et al., "Remote inspection, measurement and handling for maintenance and operation at CERN," *International Journal of Advanced Robotic Systems*, vol. 10, no. 382, pp. 1–11, 2013.

[17] N. Takeda, K. Akou, S. Kakudate et al., "Development of divertor cassette transporters for ITER," in *Proceedings of the 17th IEEE/NPSS Symposium on Fusion Engineering*, vol. 2, pp. 925–928, October 1997.

[18] P. Desbats, F. Geffard, G. Piolain, and A. Coudray, "Force-feedback teleoperation of an industrial robot in a nuclear spent fuel reprocessing plant," *Industrial Robot*, vol. 33, no. 3, pp. 178–186, 2006.

[19] F. Geffard, P. Garrec, G. Piolain et al., "TAO2000 V2 computer-assisted force feedback telemanipulators used as maintenance and production tools at the AREVA NC-La Hague fuel recycling plant," *Journal of Field Robotics*, vol. 29, no. 1, pp. 161–174, 2012.

[20] S. Sanders, "Remote operations for fusion using teleoperation," *Industrial Robot*, vol. 33, no. 3, pp. 174–177, 2006.

[21] A. Vale, D. Fonte, F. Valente, and I. Ribeiro, "Trajectory optimization for autonomous mobile robots in ITER," *Robotics and Autonomous Systems*, vol. 62, no. 6, pp. 871–888, 2014.

[22] S. Terada, H. Kobayashi, H. Sengoku et al., "Design and development of a work robot to place ATLAS SCT modules onto barrel cylinders," *Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 541, no. 1-2, pp. 144–149, 2005.

[23] H. J. Lee, J. K. Lee, B. S. Park, and J. S. Yoon, "Bridge transported servo manipulator system for remote handling tasks under a radiation environment," *Industrial Robot*, vol. 36, no. 2, pp. 165–175, 2009.

[24] J. K. Lee, B. S. Park, K. Kim, and H. D. Kim, "Design and fabrication of a servo-manipulator for use in the PRIDE facility," in *Proceedings of IEEE International Symposium on Assembly and Manufacturing (ISAM '09)*, pp. 417–421, November 2009.

[25] B. Elzendoorn, M.-D. Baar, R. Chavan et al., "Analysis of the ITER ECH Upper Port Launcher remote maintenance using virtual reality," *Fusion Engineering and Design*, vol. 84, no. 2–6, pp. 733–735, 2009.

[26] N. Takeda, S. Kakudate, M. Nakahira, K. Shibanuma, and A. Tesini, "Development of a virtual reality simulator for the ITER blanket remote handling system," *Fusion Engineering and Design*, vol. 83, no. 10–12, pp. 1837–1840, 2008.

[27] C. J. M. Heemskerk, M. R. De Baar, H. Boessenkool et al., "Extending Virtual Reality simulation of ITER maintenance operations with dynamic effects," *Fusion Engineering and Design*, vol. 86, no. 9-11, pp. 2082–2086, 2011.

[28] J. Geng, D. Zhou, C. Lv, and Z. Wang, "A modeling approach for maintenance safety evaluation in a virtual maintenance environment," *Computer Aided Design*, vol. 45, no. 5, pp. 937–949, 2013.

[29] S. Esque, J. Mattila, M. Siuko et al., "The use of digital mock-ups on the development of the Divertor Test Platform 2," *Fusion Engineering and Design*, vol. 84, no. 2–6, pp. 752–756, 2009.

[30] R. Shuff, D. Locke, and D. A. Roulet, "ITER Hot Cell process operability analysis using discrete event simulation tools,"

Fusion Engineering and Design, vol. 86, no. 9–11, pp. 1950–1953, 2011.

[31] E. Robbins, S. Sanders, A. Williams, and P. Allan, "The use of virtual reality and intelligent database systems for procedure planning, visualisation, and real-time component tracking in remote handling operations," *Fusion Engineering and Design*, vol. 84, no. 7–11, pp. 1628–1632, 2009.

[32] H.-S. Park, S.-K. Kim, K.-W. Lee, C.-H. Jung, J.-H. Park, and S.-I. Jin, "The application of visualization and simulation in a dismantling process," *Journal of Nuclear Science and Technology*, vol. 44, no. 4, pp. 649–656, 2007.

[33] H. S. Park, C. H. Choi, S. H. Kim, B. S. Park, K. H. Kim, and H. D. Kim, "Deployment analysis and remote accessibility verification for a maintenance task in a PRIDE digital mock-up," *Annals of Nuclear Energy*, vol. 38, no. 4, pp. 767–774, 2011.

[34] A. R. Selva, R. C. Castro, and V. G. F. Frías, "Design of disassembly sequences using search strategies. Application of IDA* in state diagrams," *International Journal of Production Research*, vol. 49, no. 11, pp. 3395–3403, 2011.

[35] Y. Xing, G. Chen, X. Lai, S. Jin, and J. Zhou, "Assembly sequence planning of automobile body components based on liaison graph," *Assembly Automation*, vol. 27, no. 2, pp. 157–164, 2007.

[36] J. Ko, E. Nazarian, H. Wang, and J. Abell, "An assembly decomposition model for subassembly planning considering imperfect inspection to reduce assembly defect rates," *Journal of Manufacturing Systems*, vol. 32, no. 3, pp. 412–416, 2013.

[37] G. Dini and M. Santochi, "Automated sequencing and subassembly detection in assembly planning," *CIRP Annals—Manufacturing Technology*, vol. 41, no. 1, pp. 1–4, 1992.

[38] C. Morato, K. N. Kaipa, and S. K. Gupta, "Improving assembly precedence constraint generation by utilizing motion planning and part interaction clusters," *Computer-Aided Design*, vol. 45, no. 11, pp. 1349–1364, 2013.

[39] T. G. Chen and R. B. Xiao, "Enhancing artificial bee colony algorithm with self-adaptive searching strategy and artificial immune network operators for global optimization," *The Scientific World Journal*, vol. 2014, Article ID 438260, 12 pages, 2014.

[40] H. Xu, L. Xin, H. Wang, and N. Yang, "Distribution network planning based on improved genetic algorithm," *Guangdong Electric Power*, vol. 23, no. 6, pp. 6–9, 2010.

[41] T.-G. Chen and C.-H. Ju, "A novel artificial bee colony algorithm for solving the supply chain network design under disruption scenarios," *International Journal of Computer Applications in Technology*, vol. 47, no. 2-3, pp. 289–296, 2013.

[42] C.-H. Cheng, S. C. Ho, and C.-L. Kwan, "The use of meta-heuristics for airport gate assignment," *Expert Systems with Applications*, vol. 39, no. 16, pp. 12430–12437, 2012.

[43] S. Lorpunmance and A. Sap, "An ant colony optimization for dynamic job scheduling in grid environment," *International Journal of Computer and Information Science and Engineering*, vol. 1, pp. 207–214, 2007.

[44] P. De Lit, P. Latinne, B. Rekiek, and A. Delchambre, "Assembly planning with an ordering genetic algorithm," *International Journal of Production Research*, vol. 39, no. 16, pp. 3623–3640, 2001.

[45] H.-E. Tseng, W.-P. Wang, and H.-Y. Shih, "Using memetic algorithms with guided local search to solve assembly sequence planning," *Expert Systems with Applications*, vol. 33, no. 2, pp. 451–467, 2007.

[46] H. S. Wang, Z. H. Che, and C. J. Chiang, "A hybrid genetic algorithm for multi-objective product plan selection problem with ASP and ALB," *Expert Systems with Applications*, vol. 39, no. 5, pp. 5440–5450, 2012.

[47] J.-F. Wang, J.-H. Liu, and Y.-F. Zhong, "A novel ant colony algorithm for assembly sequence planning," *The International Journal of Advanced Manufacturing Technology*, vol. 25, no. 11-12, pp. 1137–1143, 2005.

[48] H. Shan, S.-H. Zhou, and Z.-H. Sun, "Research on assembly sequence planning based on genetic simulated annealing algorithm and ant colony optimization algorithm," *Assembly Automation*, vol. 29, no. 3, pp. 249–256, 2009.

[49] Y. Wang and J. H. Liu, "Chaotic particle swarm optimization for assembly sequence planning," *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 2, pp. 212–222, 2010.

[50] M. Li, B. Wu, Y. Hu, C. Jin, and T. Shi, "A hybrid assembly sequence planning approach based on discrete particle swarm optimization and evolutionary direction operation," *The International Journal of Advanced Manufacturing Technology*, vol. 68, no. 1–4, pp. 617–630, 2013.

[51] Y. Wang and J.-H. Liu, "Chaotic particle swarm optimization for assembly sequence planning," *Robotics and Computer-Integrated Manufacturing*, vol. 26, no. 2, pp. 212–222, 2010.

[52] M. Li, B. Wu, P. Yi, C. Jin, Y. Hu, and T. Shi, "An improved discrete particle swarm optimization algorithm for high-speed trains assembly sequence planning," *Assembly Automation*, vol. 33, no. 4, pp. 360–373, 2013.

[53] G. Percoco and M. Diella, "Preliminary evaluation of artificial bee colony algorithm when applied to multi objective partial disassembly planning," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 6, no. 17, pp. 3234–3243, 2013.

[54] A. Rahimi-Vahed and A. H. Mirzaei, "A hybrid multi-objective shuffled frog-leaping algorithm for a mixed-model assembly line sequencing problem," *Computers & Industrial Engineering*, vol. 53, no. 4, pp. 642–666, 2007.

[55] A. Rahimi-Vahed, M. Dangchi, H. Rafiei, and E. Salimi, "A novel hybrid multi-objective shuffled frog-leaping algorithm for a bi-criteria permutation flow shop scheduling problem," *International Journal of Advanced Manufacturing Technology*, vol. 41, no. 11-12, pp. 1227–1239, 2009.

[56] X. Li, J.-P. Luo, M.-R. Chen, and N. Wang, "An improved shuffled frog-leaping algorithm with extremal optimisation for continuous optimisation," *Information Sciences*, vol. 192, no. 1, pp. 143–151, 2012.

[57] C. Fang and L. Wang, "An effective shuffled frog-leaping algorithm for resource-constrained project scheduling problem," *Computers & Operations Research*, vol. 39, no. 5, pp. 890–901, 2012.

[58] J.-Q. Li, Q. Pan, and S.-X. Xie, "An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems," *Applied Mathematics and Computation*, vol. 218, no. 18, pp. 9353–9371, 2012.