sustainability

MDPI

Article

Hybrid and Cooperative Strategies Using Harmony Search and Artificial Immune Systems for Solving the Nurse Rostering Problem

Suk Ho Jin [1], Ho Yeong Yun [1], Suk Jae Jeong [2,*] and Kyung Sup Kim [1,*]

1 Department of Industrial Engineering, Yonsei University, 50 Yonsei-ro, Seodaemun-gu, Seoul 03722, Korea; jinsukho@yonsei.ac.kr (S.H.J.); yhy900211@yonsei.ac.kr (H.Y.Y.)
2 Business School, Kwangwoon University, 20 Kwangwoon-ro, Nowon-gu, Seoul 01897, Korea
* Correspondence: sjjeong@kw.ac.kr (S.J.J.); kyungkim@yonsei.ac.kr (K.S.K.); Tel.: +82-2-940-5294 (S.J.J.); +82-2-2123-4012 (K.S.K.)

Received: 17 April 2017; Accepted: 20 June 2017; Published: 22 June 2017

Abstract: The nurse rostering problem is an important search problem that features many constraints. In a nurse rostering problem, these constraints are defined by processes such as maintaining work regulations, assigning nurse shifts, and considering nurse preferences. A number of approaches to address these constraints, such as penalty function methods, have been investigated in the literature. We propose two types of hybrid metaheuristic approaches for solving the nurse rostering problem, which are based on combining harmony search techniques and artificial immune systems to balance local and global searches and prevent slow convergence speeds and prematurity. The proposed algorithms are evaluated against a benchmarking dataset of nurse rostering problems; the results show that they identify better or best known solutions compared to those identified in other studies for most instances. The results also show that the combination of harmony search and artificial immune systems is better suited than using single metaheuristic or other hybridization methods for finding upper-bound solutions for nurse rostering problems and discrete optimization problems.

Keywords: nurse rostering problem; harmony search; artificial immune systems; hybridization; metaheuristics

1. Introduction

The problem of staff scheduling has been studied extensively over the past several decades [1]. It has been recognized as an important problem in academic and industrial fields. In today's fast-paced business environment, corporations have attempted to achieve two goals to gain a competitive advantage: improving customer satisfaction and reducing costs. Staff scheduling problem requires achieving both these goals. Personnel scheduling is especially complex when we consider both shift scheduling and day-off scheduling for organizations that operate seven days a week (e.g., airlines, hotels, call centers, and hospitals). Furthermore, when demand fluctuates over small intervals compared to the shift length, a generic allocation model becomes less useful for personnel scheduling, and an advanced model for allocation that includes overlapping shifts is needed. Thus, most personnel scheduling problems are non-deterministic polynomial-time-hard (NP-hard) problems for which various solution methods including mathematical models and heuristic approaches have been proposed.

Nurse rostering problems (NRPs) have been proven to be NP-hard; they are composed of many soft constraints that result in additional penalties when violated, along with a few hard constraints [2–4]. Ernst et al. [5] conducted a comprehensive review of the main research direction and solving method for such problems for facilitating companies that attempt to distribute their operations in ways that are cost effective, observant of industrial regulations, and attentive to individuals' work preferences. Cheang et al. [6] conducted a review of the literature on modeling and solution methodologies for NRPs that highlighted the specificity of the solution approaches and availability of benchmark problems for the various basic models of NRPs. Van den Bergh et al. [1] reviewed the literature on staff scheduling problems and identified various perspectives from which the existing literature could be classified and published work in the relevant fields of interest could be traced. They also identified trends and areas for future research. Ernst et al. [7] reviewed the rostering problem in specific application areas, as well as models and algorithms that have been reported in the literature for the solution to this problem; they also surveyed commonly used methods for solving the rostering problem. Burke et al. [8] described and critically evaluated solution approaches that span the interdisciplinary spectrum, from operation research techniques to artificial intelligence methods, and surveyed the strengths and weaknesses of the literature in outlining the key issues that must be addressed in future nurse rostering research.

Over the years, researchers have used various techniques to implement solutions for NRPs, including exact methods, metaheuristic approaches, and others. Exact methods have been used successfully to find complete solutions [9,10]. Unfortunately, such deterministic solutions require a great deal of computational time and resources to handle the many constraints involved. Thus, this approach is somewhat limited. Metaheuristic approaches that produce relatively good solutions within reasonable computational time frames are well known to be effective methods.

Examples of the use of metaheuristics to solve NRPs include application of genetic algorithms [11–13], simulated annealing [14], tabu search [15], and ant colony optimization [16].

In contrast to applications of a single metaheuristic, methods that combine two and more metaheuristics have also been introduced. Bai et al. [17] proposed an approach that combined a genetic algorithm and simulated annealing. In their study, simulated annealing was used as the local search method within a genetic algorithm procedure. Burke et al. [18] hybridized a steepest-descent improvement with a genetic algorithm and demonstrated that this hybridization was an adequate approach for solving NRPs. Awadallah et al. [19] proposed a hybridized approach for the application of the hill climbing optimization method to an artificial bee colony. In this approach, the process of the employed bee operator is replaced by that of the hill-climbing optimizer. The performance of the proposed method was evaluated by comparing with other hybridization approaches published in the literature.

Most previous studies on this subject have focused on solving NRPs by combining population-based metaheuristics (P-metas) for global search and either single-solution-based metaheuristics (S-metas) or local optimizers for local search [20–22]. Based on the advantage of the hybrid methods proposed in the literature [23,24], we propose a hybrid approach that involves the use of harmony search (HS) and artificial immune systems (AIS), both of which are well-known P-metas.

The advantages of a hybrid approach involving HS and AIS in solving optimization problems include the fact that HS is an emerging algorithm for swarm intelligence optimization and heuristic global search algorithms. This approach generates a new individual via cooperation among individuals, and its local searching ability is enhanced by fine-tuning the mechanism employed in HS. Although the use of HS may be suitable because it is simple, robust, and converges rapidly, it only updates when the solution generated is not better than the worst solution in the existing harmony memory (HM) pool during the current iteration.

AISs are universal optimal algorithms that impose few constraint conditions in an optimization problem. The use of AIS has yielded significant progress in many fields, such as function optimization, machine learning, pattern recognition, image disposal, and combinatorial optimization. However, AIS also has some shortcomings such as low convergence speed and prematurity. Recent studies have shown that the combination of AIS and other searching algorithms, especially random searching algorithms, can greatly improve the performance of AIS [25,26].

To the best of our knowledge, little research has been conducted on the application of AIS to the solution of NRPs, and few studies have mentioned the application of HS [27–29]. Awadallah et al. [27]

evaluated only specific instances defined in the 2010 International Nurse Rostering Competition. Their study did not investigate how HS would perform for large or complex NRP instances. To overcome this limitation, Hadwan et al. [28] used a real dataset from a large hospital in Malaysia to assess the performance of HS and evaluated nurse rostering benchmark problems using their advanced nurse rostering model (ANROM) [30].

We compared the results of previous research on ANROM with our proposed method to demonstrate the efficiency and effectiveness of the proposed method in producing high-quality solutions in shorter amounts of time.

The remainder of this paper is organized as follows. The basic explanation and definition of NRPs is presented in Section 2, and HS and AIS for NRPs are presented in Section 3. The procedure for hybridization and combination of HS and AIS in solving NRPs is presented in detail in Section 4. The computational experiments conducted and their results are discussed in Section 5. Conclusions and future research directions are presented in Section 6.

2. Problem Definition

NRPs involve producing a periodic (weekly, fortnightly, or monthly) duty roster for nursing staff that is subject to a variety of hard and soft constraints such as legal regulations, personnel policies, nurse preferences, and other requirements specific to a given hospital. In addition, a schedule avoiding difficult-to-follow shift patterns should be constructed and the work contract of each employee should be respected as much as possible. The term "work contract" refers to the agreement signed between the nurse and the hospital, which addresses requests for days or shifts on and off, working on weekends, maximum acceptable consecutive workdays, etc. Some work contract elements can be considered as legal requirements. It is worth considering the fairness of the problem from the work contracts perspective. For example, contract violations should be distributed evenly among all available nursing staff.

We studied various instances of NRPs based on the ANROM, one of the representative benchmark datasets of NRPs, and we demonstrated the superiority of the proposed algorithm using these benchmark data. ANROM, which was first implemented in a hospital in 1995, was the initial version, but the system evolved to deal with new and more complex real-world problems that appear continually. More than 40 hospitals in Belgium, some of which contain approximately 100 wards, replaced their time-consuming manual scheduling with this system. Although the problem is user-defined to a large extent, the software must be efficient in different settings. Each specific hospital ward should be able to formulate its problem within the restrictions of the model described in the following sections.

The constraints of NRPs can be divided into two classes: hard constraints and soft constraints; hard constraints are those that must always be satisfied. To address real-world hospital situations, ANROM considers the following set of hard constraints: a maximum of one assignment per shift type per day, which precludes the assignment of the same shift to a member of the ward more than once per day; and personnel requirements, which are usually expressed in terms of the minimum number of personnel required and the preferred number of personnel to meet the patients' needs. ANROM considers a high number of soft constraints, as shown in Table 1. They should preferably be satisfied, but violations can be tolerated when penalties are included in the evaluation function. The main goal of this study was to minimize the sum of the penalties that occur when soft constraints are violated and hard constraints are satisfied through experiments conducted using ANROM data. Table 1 presents the types of constraints indicated by each problem.

Table 1. Hard and soft constraints considered by ANROM instances.

Hard Constraints											
A Nurse Can Only Work One Shift Per Day, Nurse Skill Levels and Categories, Nurses' Requirements											
Soft Constraints	Instance										
	1.8.1	2.46.1	3.46.1	3.46.2	4.13.1	5.4.1	6.13.1	7.10.1	8.13.1	A.12.1	A.12.2
Minimum time between two assignments	√	√	√	√	√	√	√	√	√	√	√
Nurses workload (minimum/maximum)	√	√	√	√	√	√	√	√	√	√	√
Maximum number of consecutive working days	√	√	√	√	√	√	√	√	√	√	√
Maximum number of assignments on bank holidays	√	√	√	√	√	√	√		√	√	√
Maximum number of consecutive free days			√	√	√	√				√	√
Minimum number of consecutive free days	√	√	√	√	√	√	√		√	√	√
Maximum number of consecutive working weekends	√	√	√	√	√	√	√	√	√	√	√
Maximum number of working weekends in four weeks	√	√	√	√	√	√	√		√	√	√
Assign complete weekends	√	√	√	√	√	√	√	√	√	√	√
Assign identical shift types during the weekend	√				√	√				√	√
No night shift before a free weekend	√	√	√	√	√	√				√	√
Assign two free days after night shifts		√	√	√	√	√		√		√	√
Maximum number of assignments per day of the week	√	√	√	√	√	√	√		√	√	√
Number of consecutive shift types	√	√	√	√	√	√	√	√	√	√	√
Maximum number of assignments for each shift type	√	√	√	√	√	√	√	√	√	√	√
Maximum number of a shift type per week	√	√	√	√	√	√	√	√	√	√	√
Maximum number of hours worked	√	√	√	√	√	√	√	√	√	√	√
Minimum number of hours worked		√	√	√	√	√				√	√
Maximum number of hours per week									√	√	√
Restriction on the succession of shift types	√	√	√	√	√	√	√	√	√	√	√
Alternative skill category	√	√	√	√	√	√	√	√	√	√	√
Tutorship	√	√	√	√	√	√				√	√
People not allowed to work together	√	√	√	√	√	√				√	√
Day off	√	√	√	√			√	√	√	√	√
Day on (Requested assignments)		√	√	√							
Shift off		√	√	√			√		√	√	√
Shift on (Requested assignments)							√		√	√	√

3. Harmony Search and Artificial Immune Systems for NRPs

3.1. Harmony Search for NRPs

Harmony search, which was originally proposed by Lee and Geem [31] and Geem et al. [32], is a phenomenon-mimicking algorithm (also known as a metaheuristic algorithm) that was inspired by the improvisation process of musicians described in 2001. It has exhibited relatively good performance in various research areas such as city design, routing problem, RNA structure problem, and planetary migration. From an optimization perspective, each musician is represented by a decision variable and the best harmony achievable when playing together is the global optimum. A harmony search consists of the harmony memory (HM), which is similar to the population of a genetic algorithm; the harmony memory size (HMS), which reflects the size of the HM; and three operators, i.e., memory considering (MC), pitch adjusting (PA), and random selecting (RS), which are used to generate the new harmony.

The HS procedure consists of the following five main steps.

Step 1: Initialize the problem and parameters
Step 2: Initialize the harmony memory
Step 3: Improvise a new harmony
Step 4: Update the harmony memory
Step 5: Repeat Steps 3 to 4 until a predefined stopping condition is reached

For application of HS to NRPs, we designed HS vectors with a two-dimensional array structure. The array's columns represent individual days, and the array's rows represent individual nurses. A solution in the HM is two-dimensionally expressed as a specific day and nurse, as shown in Figure 1. The HS consists of three operators: MC, PA, and RS. Operators in the HS need to be carefully designed to avoid violating hard constraints. In the example illustrated in Figure 1, the information from two day shifts on Day 1 and Day 3 in HM 1 are first allocated to Day 1 and Day 3 of the new solution from the current HM, respectively. Here, the shift of Day 1 is made by harmony memory and considering the rate (HMCR) and pitch adjusting rate (PAR) leads to the shift of Day 3.

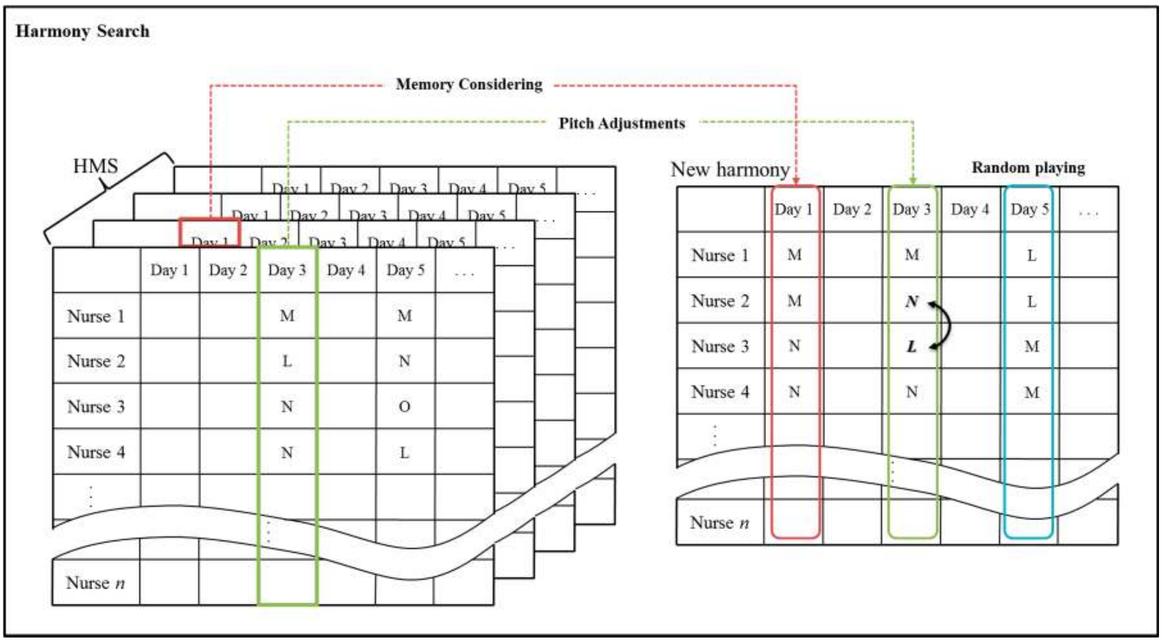


Figure 1. Application of Harmony Search (HS) operators for Nurse rostering problems (NRPs).

For the HS operator, Day 1 in HM 2 is copied to Day 1 of the new solution. For the PA operator, Day 3 shift information for the new solution is produced from Day 3 information for HM, which is randomly chosen in the HM. The shifts of randomly selected nurses are then swapped with other shifts on the same day. The frequency of swapping depends on the size of the instance. The RS operation is performed by swapping between two nurses on the same day (Day 5 in Figure 1) that are randomly selected in the new solution.

3.2. Artificial Immune Systems for NRPs

Artificial immune systems have been studied by Hunt and Cooke [33], Dasgupta [34], McCoy and Devarajan [35], Dasgupta [36], Hofmeyr and Forrest [37], and Hofmeyr [38], among others, and have been widely applied to engineering problems. These systems, well known to be efficient searching algorithms applicable for various types of combinatorial and sequence optimization problems, are inspired by theoretical immunology and observations of the principles and models of immune functions.

We utilized a clonal selection algorithm that extracts only the cloning and mutation steps of an entire AIS procedure for efficient hybridization with HS. The clonal selection algorithm is based on the principles of extraction from clonal expansion and affinity maturation [39]. The basic mechanism of clonal selection is that, when an antigen (Ag) is detected, antibodies (Abs) that become aware of this Ag will proliferate by a clonal process. The immune response is specific for each Ag.

The immune cells reproduce along with a recreating Ag until the desired results are achieved in fighting this Ag. Some of the newly cloned cells will be distinguished by plasma cells and memory. Because of the mutation procedure, the plasma cells promote genetic variation from their origins to reproduce new Abs. The memory cells are in charge of the immunologic response against future Ag attack. The best cells with the highest affinity to the Ag in the next population remain.

The following steps describe the basic procedure of the clonal selection algorithm [40,41]:

Step 1: Generate a random initial population of antibodies
Step 2: Compute the affinity of each of the antibodies
Step 3: Create new clones by cloning all cells in the population of antibodies
Step 4: Maturate cloned antibodies by mutation
Step 5: Evaluate affinity values of the clone population
Step 6: Select the best antibodies to compose the new antibody population
Step 7: Repeat Steps 3 to 6 until a predefined stopping condition is reached

Figure 2 illustrates the clonal selection mechanism used to improve global search for solving NRPs. After evaluating the affinity of each clone population, some of the antibodies with the best affinity values will clone to a degree that is inversely proportional to their affinities. Figure 3 illustrates the procedure for cloning and mutation of a single antibody. The cloned antibodies mutate to reproduce a mature clone population. To prevent generation of infeasible solutions, three swapping-based mutation operators with the same probability are applied. Case (a) in Figure 3 illustrates a "swap-shifts" situation in which the shifts of two nurses are exchanged for each of the selected days. Case (b) illustrates a "swap-nurses" situation in which two nurses are selected and then their schedules are partially or completely exchanged.

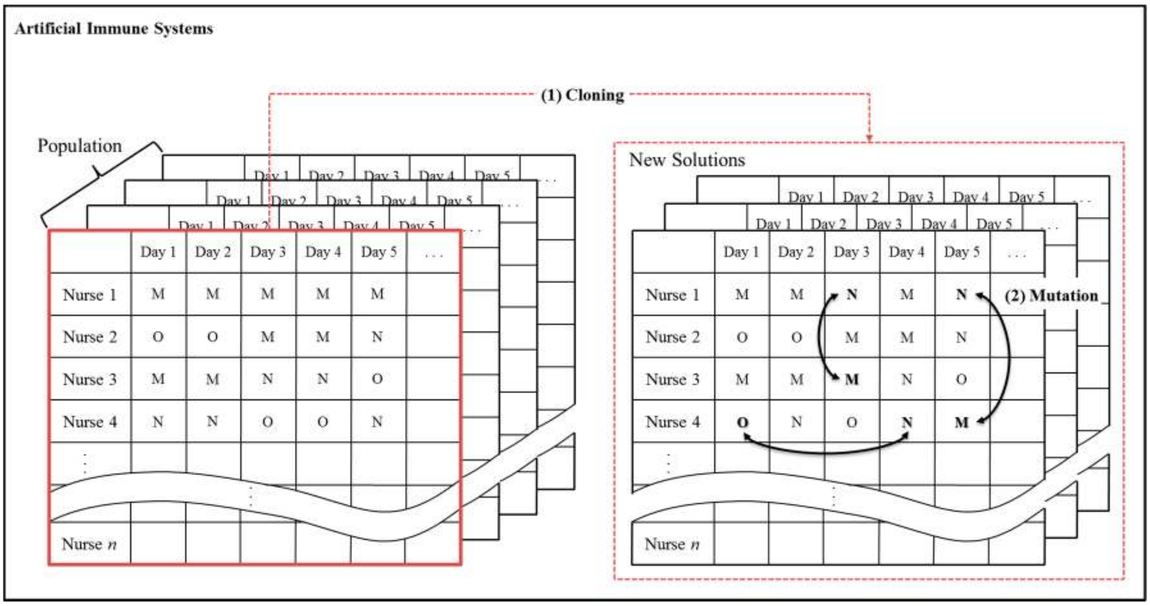


Figure 2. Application of artificial immune systems (AIS) operators for NRPs.

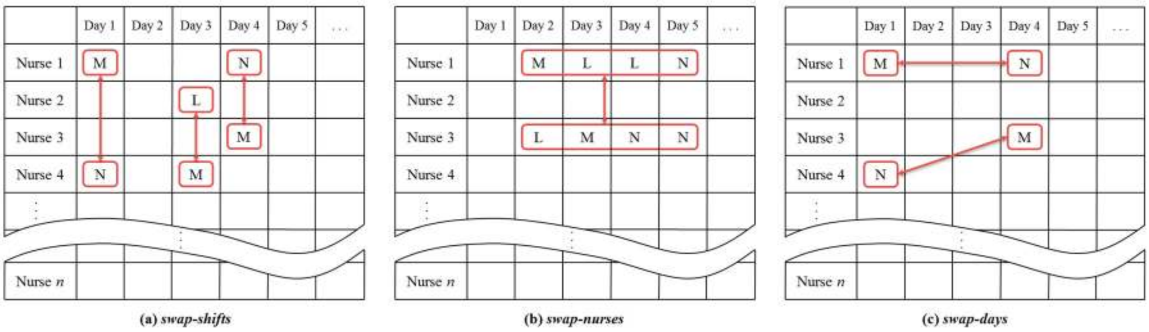


Figure 3. Application of three mutations of AIS for NRPs.

Case (c) illustrates the mutation, "swap-days", in which shifts between two differentiated days for each of the selected nurses are exchanged. Unlike the "swap-shifts" and "swap-nurses" scenarios, this scenario includes a repair process because of the possibility of violation of hard constraints. Figure 3 illustrates an example in which, if the first and fourth days of nurse 1 are interchanged, a repair would be performed by changing the shifts of nurses 3 and 4 to maintain feasibility.

4. Hybrid and Cooperative Strategies Using Harmony Search and Artificial Immune Systems

4.1. Why We Hybridize and Cooperate HS and AIS

In achieving combinatorial optimization using metaheuristic algorithms, a major concern is how to maintain the balance between two major components: diversification and intensification. These two components seem to contradict each other, but their balanced combination is crucially important to the success of obtaining a qualified solution. Proper diversification guarantees that the search in the solution space can effectively explore as many locations and regions as possible. It also ensures that the evolving system will not be trapped at biased local optima. If the diversification is too strong, it may explore too many locations in a stochastic manner and subsequently slow the convergence of the algorithm. Conversely, appropriate intensification exploits the history and experience of the search process. It also permits the convergence to be accelerated when necessary by reducing randomness and limiting diversification.

To reconcile these two conflicting components, hybrid methods using P-meta and S-meta heuristics have been used in many studies. P-metaheuristics are utilized to search among the diverse solutions in the solution space in the first and subsequent iterations. S-metaheuristics are used in the final iteration to generate solutions in the neighborhood of current ones discovered using the P-metaheuristics.

In traditional HS, a new harmony vector is generated using three rules, namely HMC, RS, and PA. A decision variable of harmony vectors is selected based on either an HMC rule with a probability of HMCR or an RS rule with a probability of 1-HMCR. A PA rule with a probability of (HMCR PAR) is then utilized to change the values of the decision variables from the HMC. Except for the PA rule, application of an HMC rule for the generation of a new harmony from the existing HMS and RS rules that generate a new harmony randomly is difficult for balancing intensification and diversification.

Many studies have attempted to develop advanced harmony searchers to maintain a balance between intensification and diversification. A typical and popular method is the improved harmony search (IHS) algorithm introduced by Mahdavi et al. [42], which employs enhanced fine-tuning characteristics and an enhanced HS convergence rate. As shown in Equations (1) and (2), the algorithm's performance is improved by dynamically increasing the pitch adjusting rate and the bandwidth (BW) as the iteration progresses.

$$\mathrm{PAR(t)} = \mathrm{PAR_{min}} + \frac{\mathrm{PAR_{max}} - \mathrm{PAR_{min}}}{\mathrm{t_{max}}} \times \mathrm{t} \tag{1}$$

$$\mathrm{BW(t)} = \mathrm{BW_{max}} \times \mathrm{e}^{\left(\ln\left(\frac{\mathrm{BW_{max}}}{\mathrm{BW_{min}}}\right) \times \frac{\mathrm{t}}{\mathrm{t_{max}}}\right)} \tag{2}$$

Although effective control of BW can be useful in balancing intensification and diversification, the characteristics of sequential optimization with main constraints such as NRPs result in the limited application of BW because of the possibility of generating worse solutions. The application of fine-tuning by PAR also has some drawbacks in early iterations in which the value of PAR is low. In spite of the critical importance of the PA rule in balancing exploitation and exploration, low values of PAR in early iterations forces premature convergence.

Highly reliable harmony search (HRHS) has been proposed by Taherinejad [43] to overcome a critical issue in IHS that may result in premature convergence in early iterations. As shown in Equation (3), HRHS guarantees diversification of good solutions that are generated in early iterations by increasing the probability of generation of neighbor solutions and vice versa. HRHS still has major drawbacks in the final iteration, where the value of PAR is close to zero and may result in stagnation in convergence of the algorithm.

$$\mathrm{PAR(t)} = \mathrm{PAR_{max}} - \frac{\mathrm{PAR_{max}} - \mathrm{PAR_{min}}}{\mathrm{t_{max}}} \times \mathrm{t} \tag{3}$$

Subsequently, effectively maintaining the balance between exploration and exploitation is difficult for dynamically changing PAR.

Another problem of HS is that the solution generated using the three rules is updated only when its value is better than the worst of the existing solutions. Thus, if the solution generated is not better, the existing HM does nothing. Even if a new harmony is better than the worst solution, only a new harmony is added to the existing HM. This is the weakness of HS in achieving the desired results from diversification. Hence, a new harmony is highly dependent on the solution characteristics of each harmony that consists of the existing HM. When premature convergence occurs, the HS method is limited in that it continually searches only local solutions. To overcome some of the limitations mentioned earlier, we propose two methods for combining HS and AIS. Determining how and when to update the existing HM is essential to the success of this approach.

4.2. How We Hybridize and Cooperate HS and AIS

The first method, "Hybrid harmony search with artificial immune systems" (HHSAIS), and its procedure are shown in Figure 4. The existing HM is updated whenever a new harmony generated by three rules is not better than the worst harmony. A new harmony not being better than the worst harmony means that the current HM may be composed of similar solutions, and this makes it extremely difficult to search the solution space thoroughly. In such a case, the insertion and support of AIS with cloning and mutation can help renew the existing HM.

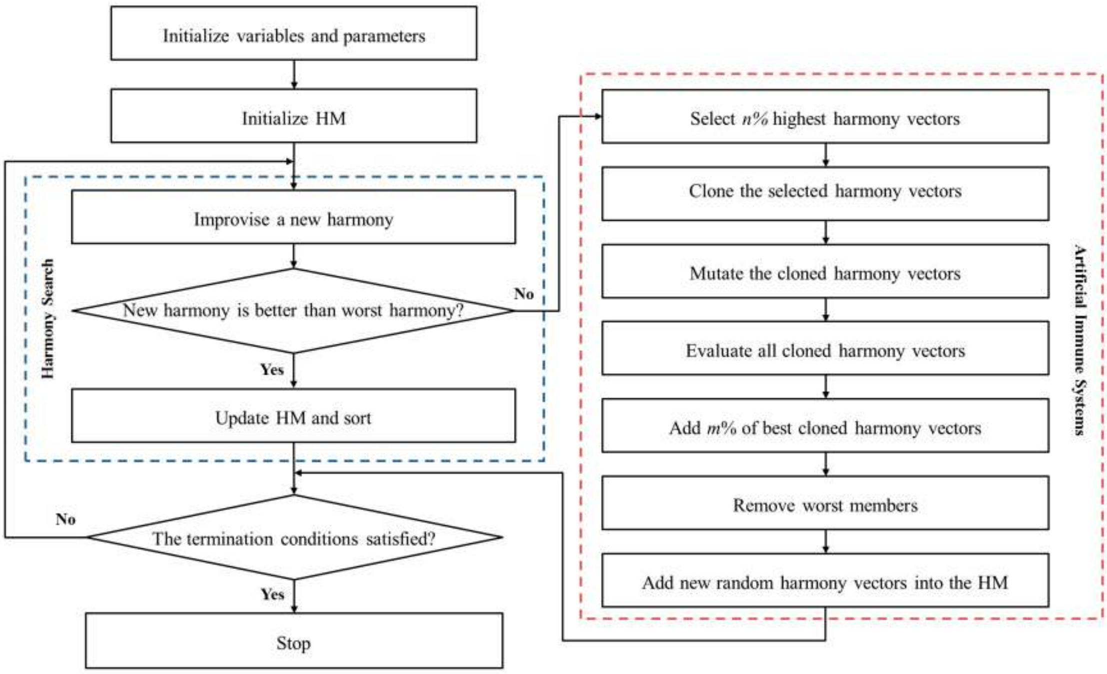


Figure 4. Application procedure for hybrid harmony search with artificial immune systems.

The second approach involves good solutions that are reproduced from both HS and AIS being handed over to the opposite population during iterations while each population of the two algorithms is maintained separately. We refer to this as the cooperative harmony search and artificial immune systems (CHSAIS) approach. CHSAIS is a way to update each existing population of HS and AIS through the injection of the opposite metaheuristics. Its core differences, compared with HHSAIS, are the sequential execution of HS and AIS and the exchange of good solution(s) generated from each other. The good solution that is generated in HS procedure is transferred to the AIS procedure to generate better solutions through the clonal and mutation operation. Conversely, the good solutions from the AIS procedure are delivered to the HM procedure to update the existing HM pool and help improve the new harmony.

By combining these two algorithms, we expect that the population of each metaheuristic during its iterations will be updated through the injection of solution(s) generated by the other party and that in the generation of a solution in the next iteration of each algorithm, the combination of the existing population and the injected good solution will have a higher probability of achieving a better solution. Figure 5 illustrates the procedure for applying CHSAIS.

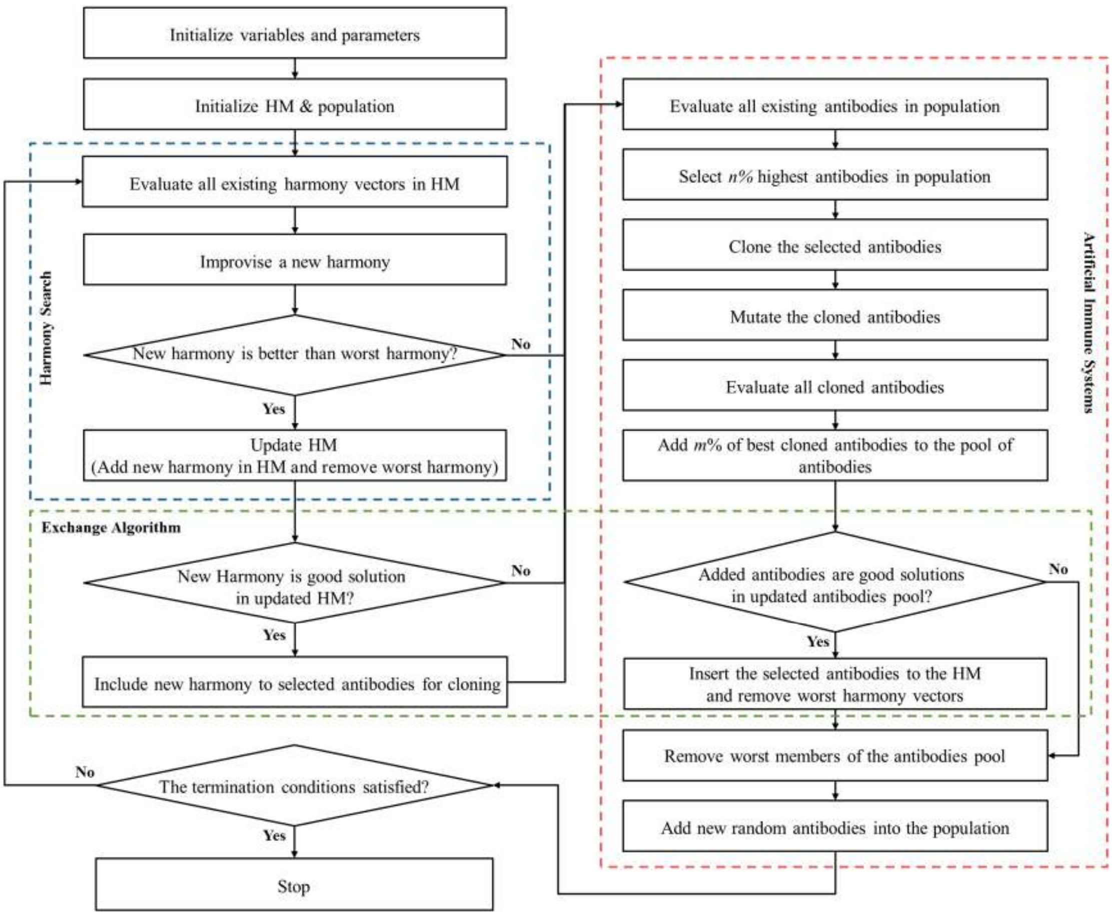


Figure 5. Application procedure for cooperative harmony search and artificial immune systems.

5. Computational Experiments

NRP benchmarking problems were defined and addressed for the purposes of validating and demonstrating the applicability of the two proposed hybrid strategies using HS and AIS. For this purpose, four algorithms were applied: HS, IHS, HHSAIS, and CHSAIS. The experiments were programmed in the C# language and carried out using a personal computer with an Intel G500 2.60-GHz processor with 4 GB of RAM and a Windows 7 operating system. As shown in Table 2, a total 18 cases—3–5 cases for each algorithm—were considered.

Cases 1 to 5 of HS were performed by changing HMCR and PAR. Cases 6 to 8 were performed by changing HMCR while performing IHS within the range of 0.1–0.9 of PAR. Cases 9 to 13 and 14 to 18 were performed by change the AIS parameters while applying HHSAIS and CHSAIS, respectively, as for Case 6. Each experimental case was replicated 30 times for each benchmarking set within the maximum number of iterations, which was set to 50,000 for all runs. The HMS, HS, and AIS population sizes were set to 10, 30, and 50, respectively, based on the number of nurses.

The experimental results are summarized in Tables 3–6 in terms of the best, mean, and worst values, standard deviations, and running times for the various cases considered. The best and mean values for each case are highlighted in bold. Table 7 shows that among the best cases for each of the four algorithms, the two proposed algorithms, HHSAIS and CHSAIS, yield results that are superior to those achieved by the HS and IHS algorithms, respectively. Figure 6a–c presents the typical solution history graphs by iteration for the four algorithms and for the BCV-5.4.1, BCV-7.10.1, and BCV-3.46.1 benchmarking sets. It can be observed that the evolution curves of the CHSAIS algorithm descend much faster and reach better solution than those of the other algorithms.

Table 2. Experimental design of proposed algorithms for different cases.

Algorithm Applications	Cases	Harmony Search		Artificial Immune Systems	
		HMCR	PAR	Threshold Proportion	New Clone Proportion
HS	Case 1	0.99	0.1	-	-
	Case 2	0.95	0.1	-	-
	Case 3	0.90	0.1	-	-
	Case 4	0.99	0.4	-	-
	Case 5	0.99	0.7	-	-
IHS	Case 6	0.99	0.1–0.9	-	-
	Case 7	0.95	0.1–0.9	-	-
	Case 8	0.90	0.1–0.9	-	-
HHSAIS	Case 9	0.99	0.1–0.9	0.3	0.3
	Case 10	0.99	0.1–0.9	0.2	0.3
	Case 11	0.99	0.1–0.9	0.1	0.3
	Case 12	0.99	0.1–0.9	0.3	0.2
	Case 13	0.99	0.1–0.9	0.3	0.1
CHSAIS	Case 14	0.99	0.1–0.9	0.3	0.3
	Case 15	0.99	0.1–0.9	0.2	0.3
	Case 16	0.99	0.1–0.9	0.1	0.3
	Case 17	0.99	0.1–0.9	0.3	0.2
	Case 18	0.99	0.1–0.9	0.3	0.1

A comparison of the results with those reported in the literature shows that the CHSAIS algorithm proposed in this paper usually yields better results than either HS and IHS alone. The results summarized in Table 8 show that CHSAIS yields competitive results in some instances. For the BCV-1.8.1, BCV-A12.1, and BCV-A12.2 cases, we obtained better results than the algorithms reported in the literature. For the BCV-4.13.1, BCV-5.4.1, BCV-7.10.1, and BCV-8.13.1 cases, CHSAIS yields the same best known results as the other methods. In the cases of BCV-2.46.1, BCV-3.46.2, and BCV-6.13.1, the results are slightly worse than the best known results. The average results for CHSAIS were compared to those reported in the literature. As the results show, CHSAIS matched the best average results obtained by other approaches in four instances and achieved better average results for six of eleven instances.

Table 3. Results of harmony search (HS) (**Bold**, optimal solutions).

Instances	Case 1					Case 2					Case 3					Case 4					Case 5				
	Best	Mean	Worst	Std.	Time	Best	Mean	Worst	Std.	Time	Best	Mean	Worst	Std.	Time	Best	Mean	Worst	Std.	Time	Best	Mean	Worst	Std.	Time
BCV-1.8.1	**252**	264.9	388	31.4	114	**252**	346.3	478	103.4	115	**252**	961.6	1348	357.5	102	**252**	263.3	422	43.1	140	**252**	**252**	252	0	131
BCV-2.46.1	2089	2259.1	2468	85.3	528	3191	3819.3	4110	226.6	444	6989	8710.9	10,310	872.6	357	1966	2189.8	2444	104.0	626	**1875**	**2091.0**	2448	116.1	697
BCV-3.46.1	3567	**3732.0**	3838	74.8	845	5623	6248.1	6542	224.8	515	13,222	14,941	16,543	869.3	401	**3536**	3752.4	3895	92.5	760	3557	3762.5	3870	95.0	733
BCV-3.46.2	1137	1616.3	1928	159.7	549	3734	4628.4	5903	431.6	510	10,056	11,380	12,845	782.1	506	1171	**1566.4**	2038	226.6	541	**1121**	1587.7	1980	196.9	539
BCV-4.13.1	**10**	**23.8**	64	20.5	165	**10**	32.6	64	21.4	150	**10**	52.4	148	40.8	122	**10**	45.9	105	38.4	161	**10**	38.6	108	34.1	161
BCV-5.4.1	**48**	**48**	48	0	60	**48**	**48**	48	0	55	**48**	**48**	48	0	54	**48**	**48**	48	0	56	**48**	**48**	48	0	55
BCV-6.13.1	858	880.0	949	35.1	154	840	987.0	1117	72.2	141	916	2305.5	2772	552.4	115	816	871.7	911	41.9	151	**810**	**861.9**	908	43.4	151
BCV-7.10.1	**381**	**381**	381	0	162	**381**	475.3	607	103.4	164	877	1200.4	1477	177.8	146	**381**	398	551	51.9	200	**381**	**381**	381	0	187
BCV-8.13.1	**148**	**170.0**	239	35.1	171	160	296.9	425	68.5	155	1058	1790.0	2128	253.8	127	**148**	203.3	243	42.4	167	**148**	200.1	246	43.1	167
BCV-A.12.1	2120	2366.8	2630	165.2	342	2803	3640.4	3984	346.2	319	6260	7602.9	9979	985.9	260	2078	2255.5	2663	148.9	450	**1887**	**2101.2**	2468	135.4	502
BCV-A.12.2	2632	2892.9	3254	186.1	414	3326	4204.6	4822	342.5	326	6757	8107.4	10479	1015.8	274	2582	2800.7	3297	176.0	490	**2381**	**2646.1**	2968	147.3	542

Table 4. Results of improved harmony search (IHS) (**Bold**, optimal solutions).

Instances	Case 6					Case 7					Case 8				
	Best	Mean	Worst	Std.	Time	Best	Mean	Worst	Std.	Time	Best	Mean	Worst	Std.	Time
BCV-1.8.1	**252**	**252**	252	0	124	**252**	309.2	462	89.3	110	**252**	830.7	1336	445.4	88
BCV-2.46.1	**1701**	**1947.1**	2050	89.9	753	2992	3768.9	4043	227.1	636	5801	7184.7	8184	506.8	509
BCV-3.46.1	**3478**	**3723.0**	3827	89.3	873	5865	6643.8	6916	227.5	447	13,530	14,885.0	15,417	471.4	381
BCV-3.46.2	**1247**	**1561.5**	2177	206.4	508	3872	4555.6	5749	387.8	473	9877	11,410.1	13,129	625.8	511
BCV-4.13.1	**10**	**21.9**	64	19.1	199	**10**	28.1	84	24.1	186	**10**	27.5	52	16.4	153
BCV-5.4.1	**48**	**48**	48	0	65	**48**	**48**	48	0	63	**48**	**48**	48	0	62
BCV-6.13.1	**806**	**840.1**	897	37.3	186	**806**	927.8	997	51.9	174	889	1963.0	2320	368.7	143
BCV-7.10.1	**381**	**381**	381	0	178	**381**	438.2	591	89.3	157	856	1205.0	1465	158.5	126
BCV-8.13.1	**148**	**182.1**	239	37.3	206	158	272.9	339	46.3	193	1017	1410.2	1676	175.9	158
BCV-A.12.1	**1772**	**1900.5**	2362	105.4	501	2550	3614.6	4012	316.9	447	5702	7025.8	8075	462.9	360
BCV-A.12.2	**2272**	**2402.9**	2862	104.7	537	3067	4117.6	4512	314.4	463	6348	7533.5	8575	448.3	363

Table 5. Results of hybrid harmony search with artificial immune systems (**Bold**, optimal solutions).

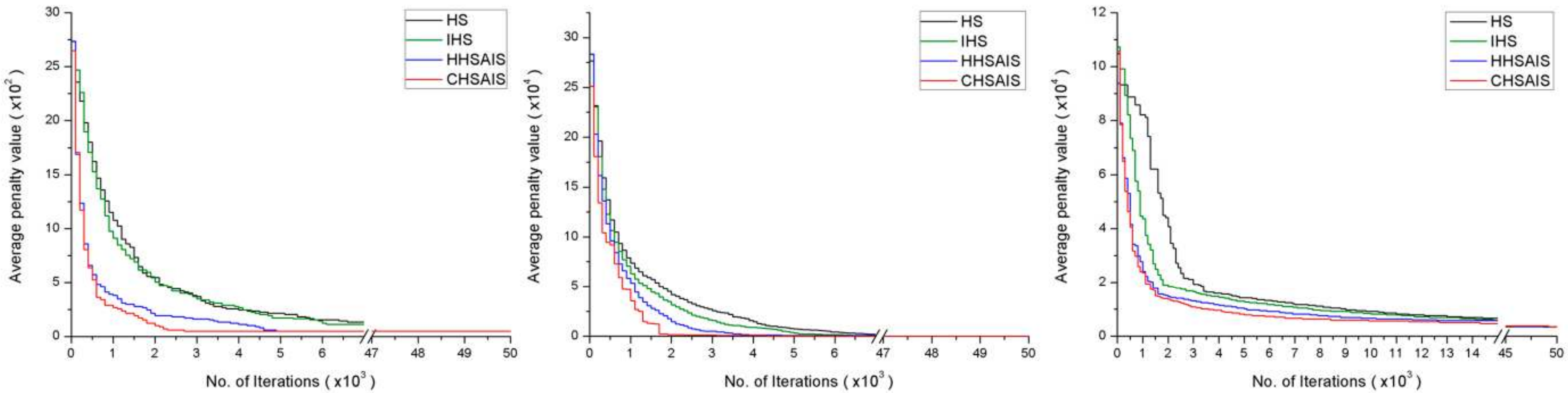
Instances	Case 9					Case 10					Case 11					Case 12					Case 13				
	Best	Mean	Worst	Std.	Time	Best	Mean	Worst	Std.	Time	Best	Mean	Worst	Std.	Time	Best	Mean	Worst	Std.	Time	Best	Mean	Worst	Std.	Time
BCV-1.8.1	**252**	**252**	252	0	268	**252**	**252**	252	0	282	**252**	**252**	252	0	255	**252**	**252**	252	0	258	**252**	**252**	252	0	261
BCV-2.46.1	1723	1868.7	2440	122.0	762	**1688**	1839.4	2374	116.5	761	1734	1903.6	2378	108.8	718	1693	**1825.3**	2469	135.5	729	1720	1852.1	2687	168.1	738
BCV-3.46.1	3580	3830.3	3948	94.8	854	3627	3802.1	3921	93.5	865	3646	3870.3	3993	104.1	884	3660	**3786.7**	3924	79.6	876	**3540**	3807.2	3949	99.1	837
BCV-3.46.2	954	**971.1**	1025	12.4	1771	**947**	975.8	1058	18.9	993	950	979.7	1029	15.8	1050	947	976.3	1014	15.4	1233	964	1001.9	1132	29.4	1360
BCV-4.13.1	**10**	13	38	7.5	284	**10**	13.2	36	7.6	281	**10**	13.2	36	7.3	274	**10**	**12.9**	38	7.3	283	**10**	13.2	34	6.8	276
BCV-5.4.1	**48**	**48**	48	0	160	**48**	**48**	48	0	152	**48**	**48**	48	0	149	**48**	**48**	48	0	158	**48**	**48**	48	0	152
BCV-6.13.1	824	852.8	883	26.5	266	814	841.2	887	32.5	264	**794**	**828.2**	882	39.8	257	798	840.9	883	40.7	264	805	839.6	916	23.7	260
BCV-7.10.1	**381**	**381**	381	0	383	**381**	**381**	381	0	403	**381**	**381**	381	0	364	**381**	**381**	381	0	368	**381**	**381**	381	0	372
BCV-8.13.1	**148**	196.5	239	40.6	294	**148**	187.3	243	41.2	292	**148**	183.9	238	40.1	284	**148**	194.9	239	43.0	293	**148**	**170.1**	233	32.1	286
BCV-A.12.1	1659	1912.9	2316	115.7	584	1695	**1843.7**	2174	88.5	576	1710	1948.9	2267	120.3	544	1752	1883.8	2181	93.9	551	**1632**	1902.7	2199	109.6	561
BCV-A.12.2	2200	2417.1	2816	112.0	552	2056	**2341.9**	2674	99.5	561	**2008**	2445.0	2767	137.9	591	2165	2383.8	2681	98.1	578	2188	2408.6	2699	101.6	537

Table 6. Results of cooperative harmony search and artificial immune systems (**Bold**, optimal solutions).

Instances	Case 14					Case 15					Case 16					Case 17					Case 18				
	Best	Mean	Worst	Std.	Time	Best	Mean	Worst	Std.	Time	Best	Mean	Worst	Std.	Time	Best	Mean	Worst	Std.	Time	Best	Mean	Worst	Std.	Time
BCV-1.8.1	**252**	**252**	252	0	278	**252**	**252**	252	0	254	**252**	**252**	252	0	255	**252**	**252**	252	0	254	**252**	**252**	252	0	263
BCV-2.46.1	1618	1704.9	1877	50.7	733	1621	1713.5	1782	49.4	731	1601	1690.5	1800	57.2	719	**1593**	**1686.6**	1791	45.2	752	1607	1726.3	1977	73.2	763
BCV-3.46.1	3429	3603.4	3705	71.5	902	3490	3617.7	3725	70.8	857	3345	3584.6	3693	74.7	834	**3312**	**3492.4**	3654	91.1	902	3430	3612.2	3753	81.9	885
BCV-3.46.2	933	943.2	962	6.1	1826	925	935.9	945	5.7	1117	920	944.9	965	10.0	957	**902**	**913.3**	928	9.3	1356	931	955.1	977	12.9	1896
BCV-4.13.1	**10**	10.8	18	2.4	271	**10**	10.4	18	1.6	277	**10**	10.5	14	1.2	274	**10**	**10**	11	0.2	276	**10**	10.1	11	0.3	275
BCV-5.4.1	**48**	**48**	48	0	148	**48**	**48**	48	0	145	**48**	**48**	48	0	147	**48**	**48**	48	0	159	**48**	**48**	48	0	173
BCV-6.13.1	800	820.5	879	33.9	256	803	831.2	885	37.2	262	798	828.3	885	38.6	259	**792**	**808.8**	877	27.4	261	797	835.4	884	39.7	260
BCV-7.10.1	**381**	**381**	381	0	397	**381**	**381**	381	0	363	**381**	**381**	381	0	364	**381**	**381**	381	0	363	**381**	**381**	381	0	375
BCV-8.13.1	**148**	174.0	235	35.5	281	**148**	182.6	241	40.9	287	**148**	183.2	241	39.5	284	**148**	**165.7**	234	29.9	286	**148**	190.4	240	40.7	285
BCV-A.12.1	1609	1778.5	1882	70.8	534	1574	1764.7	1873	83.0	514	1558	1732.0	1854	73.1	518	**1491**	**1659.9**	1787	81.9	533	1606	1782.1	1910	76.3	588
BCV-A.12.2	2109	2245.1	2382	79.1	531	2074	2274.7	2374	89.2	542	1998	2209.4	2316	72.8	523	**1998**	**2162.9**	2287	82.5	576	2098	2285.9	2410	80.1	607

Table 7. Comparison of results by representative cases (**Bold**, optimal solutions).

Instances	Case 5 (HS)			Case 6 (IHS)			Case 12 (HHSAIS)			Case 17 (CHSAIS)		
	Best	Mean	Time	Best	Mean	Time	Best	Mean	Time	Best	Mean	Time
BCV-1.8.1	**252**	**252**	131	**252**	**252**	124	**252**	**252**	258	**252**	**252**	254
BCV-2.46.1	1875	2091	697	1701	1947	753	1693	1825.3	729	**1593**	**1686.6**	752
BCV-3.46.1	3557	3763	733	3478	3723	873	3660	3786.7	876	**3312**	**3492.4**	902
BCV-3.46.2	1121	1588	539	1247	1562	508	947	976.33	1233	**902**	**913.3**	1356
BCV-4.13.1	**10**	38.6	161	**10**	21.9	199	**10**	12.867	283	**10**	**10**	276
BCV-5.4.1	**48**	**48**	55	**48**	**48**	65	**48**	**48**	158	**48**	**48**	159
BCV-6.13.1	810	861.9	151	806	840.1	186	798	840.93	264	**792**	**808.8**	261
BCV-7.10.1	**381**	**381**	187	**381**	**381**	178	**381**	**381**	368	**381**	**381**	363
BCV-8.13.1	**148**	200.1	167	**148**	182.1	206	**148**	194.87	293	**148**	**165.7**	286
BCV-A.12.1	1887	2101	502	1772	1900	501	1752	1883.8	551	**1491**	**1659.9**	533
BCV-A.12.2	2381	2646	542	2272	2403	537	2165	2383.8	578	**1998**	**2162.9**	576



(*a*) BCV 5.4.1 (*b*) BCV 7.10.1 (*c*) BCV 3.46.1

Figure 6. Evolution of average penalty value by algorithm.

Table 8. Comparison of CHSAIS results and other metaheuristic methods (**Bold**, optimal solutions).

Instances	CHSAIS (Proposed)		A1		A2		A3		A4	
	Best	Mean	Best	Mean	Best	Mean	Best	Mean	Best	Mean
BCV-1.8.1	**252**	**252**	270	272.6	272	288	**252**	253	256	261
BCV-2.46.1	1593	1686.6	1612	1630.2	**1572**	1587	**1572**	**1572**	**1572**	**1572**
BCV-3.46.1	**3312**	3492.4	3380	3391.6	3565	3631	3351	**3357**	3364	3387
BCV-3.46.2	902	913.3	905	909.8	908	911	**894**	**894**	900	902
BCV-4.13.1	**10**	**10**	11	12	12	45	**10**	**10**	**10**	**10**
BCV-5.4.1	**48**	**48**	**48**	**48**	**48**	136	**48**	**48**	**48**	**48**
BCV-6.13.1	792	808.8	796	869	964	1060	**784**	**784**	875	930
BCV-7.10.1	**381**	**381**	386	411.4	**381**	387	**381**	382	**381**	**381**
BCV-8.13.1	**148**	165.7	158	164.4	**148**	149	**148**	**148**	**148**	**148**
BCV-A.12.1	**1491**	**1659.9**	2210	2491.8	1880	2239	1600	1733	1640	1843
BCV-A.12.2	**1998**	**2162.9**	**1998**	2223.6	2528	2812	2180	2321	2465	2562

A1, Harmony search by Hadwan et al. [28]; A2, Scatter search using hill climber by Burke et al. [44]; A3, Scatter search using variable-depth search by Burke et al. [44]; A4, Memetic algorithm by Burke et al. [18].

In addition, we did experiments with INRC 2010 instances [45] for comparing among our proposed and previous works. Table 9 show the gaps among the results of our proposed and previous works published previously. The results lead that our proposed is superior over the other ones in solving the NRP.

Table 9. Comparison of CHSAIS results and other meta-heuristic methods on benchmark NRPs (**Bold**, best known solutions).

Instances	Best Known	CHSAIS		B1 Best	B2 Best
		Best	Mean		
Sprint_Early01	**56**	**56**	58.5	**56**	58
Sprint_Early02	**58**	**58**	60.5	**58**	60
Sprint_Early03	**51**	**51**	53.7	**51**	53
Sprint_Early04	**59**	**59**	61.8	**59**	62
Sprint_Early05	**58**	**58**	61.0	**58**	59
Medium_Early01	**240**	244	247.3	245	270
Medium_Early02	**240**	241	247.4	245	275
Medium_Early03	**236**	238	243.6	242	265
Medium_Early04	**237**	242	244.4	240	263
Medium_Early05	**303**	308	311.1	308	334
Long_Early01	**197**	**197**	205.1	**197**	256
Long_Early02	**219**	**219**	226.4	229	299
Long_Early03	**240**	242	249.3	**240**	286
Long_Early04	**303**	**303**	311.0	**303**	356
Long_Early05	**284**	**284**	291.9	**284**	337

B1, Hybrid Artificial Bee Colony Algorithms by Awadallah et al. [19]; B2, Global best harmony search by Awadallah et al. [29].

Table 10 shows the comparative results between CHSAIS and general GA. As you can see, despite the various changes of parameters of GA, the results lead to our algorithm being superior to all cases of GA. Because of having randomness during iterations, the corresponding t-test was conducted to assess the statistical significance is determined by the p-value.

Table 10. Comparative results between CHSAIS and Genetic Algorithms (**Bold**, optimal solutions).

Instances	CHSAIS (Proposed)		GA-1 CRate: 0.99/MRate: 0.1			GA-2 CRate: 0.95/MRate: 0.1			GA-3 CRate: 0.90/MRate: 0.1			GA-4 CRate: 0.99/MRate: 0.4			GA-5 CRate: 0.99/MRate: 0.7		
	Best	Mean	Best	Mean	Significance Probability	Best	Mean	Significance Probability	Best	Mean	Significance Probability	Best	Mean	Significance Probability	Best	Mean	Significance Probability
BCV-1.8.1	**252**	**252**	**252**	282.6	0.000275	**252**	381.1	0.000004	285	1090.9	0.000000	**252**	281.8	0.001488	**252**	273.1	0.000068
BCV-2.46.1	**1593**	**1686.6**	2258	2786.5	0.000000	3567	4581.9	0.000000	7794	10745.1	0.000000	2194	2698.5	0.000000	1999	2548.6	0.000000
BCV-3.46.1	**3312**	**3492.4**	3674	4523	0.000002	6166	7348.9	0.000000	14,635	17,718.3	0.000000	3605	4479.9	0.000001	3478	4547.2	0.000000
BCV-3.46.2	**902**	**913.3**	1424	1984.3	0.000000	3872	5560.9	0.000000	11,464	14,124.4	0.000000	1446	1884.9	0.000000	1350	1894.4	0.000000
BCV-4.13.1	**10**	**10**	**10**	24	0.000790	**10**	32.7	0.000003	**10**	52.5	0.000004	**10**	46.4	0.000012	**10**	36.5	0.000032
BCV-5.4.1	**48**	**48**	**48**	**48**	-	**48**	**48**	-	**48**	**48**	-	**48**	**48**	-	**48**	**48**	-
BCV-6.13.1	**792**	**808.8**	865	955.8	0.000000	933	1068.1	0.000000	976	2515.3	0.000000	837	961.3	0.000000	821	943.5	0.000000
BCV-7.10.1	**381**	**381**	**381**	395.1	0.000001	**381**	492.3	0.000003	1130	1254.4	0.000000	**381**	406.2	0.021185	**381**	385	0.019940
BCV-8.13.1	**148**	**165.7**	228	483	0.000000	267	357.9	0.003998	1058	1790	0.000000	155	209.5	0.000000	149	201.9	0.000004
BCV-A.12.1	**1491**	**1659.9**	2310	2830.4	0.000000	2925	4456.8	0.000000	6723	9098.9	0.000000	2146	2822.1	0.000000	2042	2564.4	0.000000
BCV-A.12.2	**1998**	**2162.9**	2705	3404.6	0.000000	3661.0	5124.5	0.000000	7124.0	9757.4	0.000000	2779.0	3496	0.000000	2534	3251.4	0.000000

Crate, Crossover Rate; MRate, Mutation Rate.

6. Conclusions

In this paper, two strategies in applying two population based metaheuristics, named HHSAIS and CHSAIS, is designed. We approach HHSAIS to solve NRP because, even though the new solution generated from HS procedure is not better than the worst existing HM, we expect that AIS can make it a better solution by searching neighbor solutions through cloning and mutation operator. Unlike local search, which provides pure exploitation without exploration, the hybridization of HS and AIS can be better harmony in dealing with the NPR's nature of the search space for the highly constrained optimization problem.

The proposed methods were applied in combination with general HS procedures to newly updated HM and sequential executions of HS and AIS, with solution exchanges. We evaluated the various methods in terms of instances of NRP benchmarking collected from ANROM.

The experimental results using HHSAIS demonstrate justification of our beliefs. Second is CHSAIS, which is cooperative of HS and AIS. In this approach, both algorithms operate individually and solutions generated from each algorithm are exchanged in the opposite population every iteration. Through this approach, we expect that, even if the computation time is longer than HS alone and CHSAIS, it has significant advantages as the solutions are swapped between each other from opposite algorithm, thus can actively explore different search space regions. Clearly, the CHSAIS matched the best average results obtained by other approaches in four instances and achieved better average results for six of eleven instances by other comparative methods. The fact that the proposed CHSAIS has the ability to explore the solution search space of the NPR in different ways to generate desired solutions could be an exploratory for researchers in the future.

In future research, we will test the superiority of our algorithm by applying it to real hospital data and we will attempt to generalize our algorithm to solve various combinatorial and sequential optimization problems.

Author Contributions: S.H. Jin developed the model and wrote the majority of the manuscript. H.Y. Yun performed the experiments and analyzed data. S.J. Jeong wrote the experimental section of the paper. K.S. Kim developed the overall idea and the basic outline of the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Van den Bergh, J.; Beliën, J.; De Bruecker, P.; Demeulemeester, E.; De Boeck, L. Personnel scheduling: A literature review. *Eur. J. Oper. Res.* **2013**, *226*, 367–385. [CrossRef]
2. Karp, R.M. Reducibility among combinatorial problems. In *Complexity of Computer Computations*; Springer: New York, NY, USA, 1972; pp. 85–103.
3. Osogami, T.; Imai, H. Classification of various neighborhood operations for the nurse scheduling problem. In Proceedings of the International Symposium on Algorithms and Computation, Taipei, Taiwan, 18–20 December 2000; pp. 72–83.
4. Valouxis, C.; Gogos, C.; Goulas, G.; Alefragis, P.; Housos, E. A systematic two phase approach for the nurse rostering problem. *Eur. J. Oper. Res.* **2012**, *219*, 425–433. [CrossRef]
5. Ernst, A.T.; Jiang, H.; Krishnamoorthy, M.; Owens, B.; Sier, D. An annotated bibliography of personnel scheduling and rostering. *Ann. Oper. Res.* **2004**, *127*, 21–144. [CrossRef]
6. Cheang, B.; Li, H.; Lim, A.; Rodrigues, B. Nurse rostering problems—A bibliographic survey. *Eur. J. Oper. Res.* **2003**, *151*, 447–460. [CrossRef]
7. Ernst, A.T.; Jiang, H.; Krishnamoorthy, M.; Sier, D. Staff scheduling and rostering: A review of applications, methods and models. *Eur. J. Oper. Res.* **2004**, *153*, 3–27. [CrossRef]
8. Burke, E.K.; De Causmaecker, P.; Berghe, G.V.; Van Landeghem, H. The state of the art of nurse rostering. *J. Sched.* **2004**, *7*, 441–499. [CrossRef]
9. Santos, H.G.; Toffolo, T.A.; Gomes, R.A.; Ribas, S. Integer programming techniques for the nurse rostering problem. *Ann. Oper. Res.* **2016**, *239*, 225–251. [CrossRef]

10. Rahimian, E.; Akartunalı, K.; Levine, J. A hybrid integer programming and variable neighbourhood search algorithm to solve nurse rostering problems. *Eur. J. Oper. Res.* **2017**, *258*, 411–423. [CrossRef]
11. Cai, X.; Li, K. A genetic algorithm for scheduling staff of mixed skills under multi-criteria. *Eur. J. Oper. Res.* **2000**, *125*, 359–369. [CrossRef]
12. Aickelin, U.; Dowsland, K.A. An indirect genetic algorithm for a nurse-scheduling problem. *Comput. Oper. Res.* **2004**, *31*, 761–778. [CrossRef]
13. Tsai, C.-C.; Li, S.H. A two-stage modeling with genetic algorithms for the nurse scheduling problem. *Expert Syst. Appl.* **2009**, *36*, 9506–9512. [CrossRef]
14. Brusco, M.J.; Jacobs, L.W. Cost analysis of alternative formulations for personnel scheduling in continuously operating organizations. *Eur. J. Oper. Res.* **1995**, *86*, 249–261. [CrossRef]
15. Burke, E.; De Causmaecker, P.; Berghe, G.V. A hybrid tabu search algorithm for the nurse rostering problem. In Proceedings of the Asia-Pacific Conference on Simulated Evolution and Learning, Canberra, Australia, 24–27 November 1998; pp. 187–194.
16. Gutjahr, W.J.; Rauner, M.S. An aco algorithm for a dynamic regional nurse-scheduling problem in Austria. *Comput. Oper. Res.* **2007**, *34*, 642–666. [CrossRef]
17. Bai, R.; Burke, E.K.; Kendall, G.; Li, J.; McCollum, B. A hybrid evolutionary approach to the nurse rostering problem. *IEEE Trans. Evol. Comput.* **2010**, *14*, 580–590. [CrossRef]
18. Burke, E.; Cowling, P.; De Causmaecker, P.; Berghe, G.V. A memetic approach to the nurse rostering problem. *Appl. Intell.* **2001**, *15*, 199–214. [CrossRef]
19. Awadallah, M.A.; Bolaji, A.L.; Al-Betar, M.A. A hybrid artificial bee colony for a nurse rostering problem. *Appl. Soft Comput.* **2015**, *35*, 726–739. [CrossRef]
20. Gogna, A.; Tayal, A. Metaheuristics: Review and application. *J. Exp. Theor. Artif. Intell.* **2013**, *25*, 503–526. [CrossRef]
21. BoussaïD, I.; Lepagnot, J.; Siarry, P. A survey on optimization metaheuristics. *Inf. Sci.* **2013**, *237*, 82–117. [CrossRef]
22. Liefooghe, A.; Humeau, J.; Mesmoudi, S.; Jourdan, L.; Talbi, E.-G. On dominance-based multiobjective local search: Design, implementation and experimental analysis on scheduling and traveling salesman problems. *J. Heuristics* **2012**, *18*, 317–352. [CrossRef]
23. Sheikholeslami, R.; Khalili, B.G.; Sadollah, A.; Kim, J. Optimization of reinforced concrete retaining walls via hybrid firefly algorithm with upper bound strategy. *KSCE J. Civ. Eng.* **2016**, *20*, 2428–2438. [CrossRef]
24. Pahnehkolaei, S.M.A.; Alfi, A.; Sadollah, A.; Kim, J.H. Gradient-based water cycle algorithm with evaporation rate applied to chaos suppression. *Appl. Soft Comput.* **2016**, *53*, 420–440. [CrossRef]
25. Abdollahpour, S.; Rezaeian, J. Minimizing makespan for flow shop scheduling problem with intermediate buffers by using hybrid approach of artificial immune system. *Appl. Soft Comput.* **2015**, *28*, 44–56. [CrossRef]
26. Mahapatra, P.K.; Ganguli, S.; Kumar, A. A hybrid particle swarm optimization and artificial immune system algorithm for image enhancement. *Soft Comput.* **2015**, *19*, 2101–2109. [CrossRef]
27. Awadallah, M.A.; Khader, A.T.; Al-Betar, M.A.; Bolaji, A.L. Nurse rostering using modified harmony search algorithm. In Proceedings of the International Conference on Swarm, Evolutionary, and Memetic Computing, Visakhapatnam, India, 19–21 December 2011; pp. 27–37.
28. Hadwan, M.; Ayob, M.; Sabar, N.R.; Qu, R. A harmony search algorithm for nurse rostering problems. *Inf. Sci.* **2013**, *233*, 126–140. [CrossRef]
29. Awadallah, M.A.; Khader, A.T.; Al-Betar, M.A.; Bolaji, A.L. Global best harmony search with a new pitch adjustment designed for nurse rostering. *J. King Saud Univ. Comput. Inf. Sci.* **2013**, *25*, 145–162. [CrossRef]
30. Burke, E.K.; Curtois, T.; Qu, R.; Vanden Berghe, G. *Problem Model for Nurse Rostering Benchmark Instances*; Technical Report; ASAP, School of Computer Science, University of Nottingham Jubilee Campus: Nottingham, UK, 2008.
31. Lee, K.S.; Geem, Z.W. A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice. *Comput. Methods Appl. Mech. Eng.* **2005**, *194*, 3902–3933. [CrossRef]
32. Geem, Z.W.; Kim, J.H.; Loganathan, G. A new heuristic optimization algorithm: Harmony search. *Simulation* **2001**, *76*, 60–68. [CrossRef]
33. Hunt, J.E.; Cooke, D.E. Learning using an artificial immune system. *J. Netw. Comput. Appl.* **1996**, *19*, 189–212. [CrossRef]

34. Dasgupta, D. Artificial neural networks and artificial immune systems: Similarities and differences. In Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997; pp. 873–878.
35. McCoy, D.F.; Devarajan, V. Artificial immune systems and aerial image segmentation. In Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997; pp. 867–872.
36. Dasgupta, D. Immunity-based intrusion detection system: A general framework. In Proceedings of the 22nd National Information Systems Security Conference (NISSC), Arlington, VA, USA, 18–21 October 1999; pp. 147–160.
37. Forrest, S.; Hofmeyr, S. Immunity by design: An artificial immune system. In Proceedings of the Genetic and Evolutionary Computation Conference, Orlando, FL, USA, 13–17 July 1999; pp. 1289–1296.
38. Hofmeyr, S.A. An interpretative introduction to the immune system. In *Design Principles for the Immune System and Other Distributed Autonomous Systems*; Oxford University Press: Oxford, UK, 2001; pp. 28–36.
39. de Castro, L.N.; Von Zuben, F.J. Learning and optimization using the clonal selection principle. *IEEE Trans. Evol. Comput.* **2002**, *6*, 239–251. [CrossRef]
40. De Castro, L.N.; Timmis, J. *Artificial Immune Systems: A New Computational Intelligence Approach*; Springer: London, UK, 2002.
41. De Castro, L.N.; Timmis, J. Artificial immune systems: A novel paradigm to pattern recognition. In *Artificial Neural Networks in Pattern Recognition*; University of the West of Scotland: Paisley, UK, 2002; pp. 67–84.
42. Mahdavi, M.; Fesanghary, M.; Damangir, E. An improved harmony search algorithm for solving optimization problems. *Appl. Math. Comput.* **2007**, *188*, 1567–1579. [CrossRef]
43. Taherinejad, N. Highly reliable harmony search algorithm. In Proceedings of the 2009 European Conference on Circuit Theory and Design, Antalya, Turkey, 23–27 August 2009; pp. 818–822.
44. Burke, E.K.; Curtois, T.; Qu, R.; Berghe, G.V. A scatter search methodology for the nurse rostering problem. *J. Oper. Res. Soc.* **2010**, *61*, 1667–1679. [CrossRef]
45. Haspeslagh, S.; De Causmaecker, P.; Stolevik, M.; Schaerf, A. First International Nurse Rostering Competition 2010. In Proceedings of the 8th International Conference on the Practice and Theory of Automated Timetabling (PATAT), Belfast, UK, 10–13 August 2010; pp. 498–501.