International Journal of Innovative Computing, Information and Control
Volume **9**, Number **5**, May **2013**
ICIC International ©2013 ISSN 1349-4198
pp. 2043–2063

CYBER SWARM ALGORITHMS FOR MULTI-OBJECTIVE NURSE ROSTERING PROBLEM

Peng-Yeng Yin and Ya-Tzu Chiang
Department of Information Management
National Chi Nan University
No. 1, University Rd., Puli, Nantou County 54561, Taiwan
{ pyyin; s96213541 }@ncnu.edu.tw

Received February 2012; revised June 2012

Abstract. *It is time-consuming to determine the nurse rostering using traditional human-involved manner in order to account for administrative operations, business benefits, governmental regulations, and nurse requests. Moreover, the objectives cannot be measured quantitatively even when the nurse rostering is generated after a lengthy manual process. This paper presents a multiobjective optimization method based on the cyber swarm algorithm (CSA), one of the most successful variations of particle swarm optimization, to solve the nurse rostering problem with multiple objectives and a set of hard constraints. The proposed method incorporates salient features from particle swarm optimization, adaptive memory programming, scatter search and path relinking to create advantages. The experimental results on two test datasets demonstrate that the proposed method outperforms several state-of-the-art multiobjective evolutionary algorithms in obtaining results with better convergence and diversity performances.*
Keywords: Cyber swarm algorithm, Scatter search, Path relinking, Multiobjective optimisation, Nurse rostering

1. **Introduction.** Staff scheduling is a central problem in many fields of business, such as airline crew pairing, call center scheduling [1], and nurse rostering [2], to name just a few. A good schedule can not only increase monetary benefit to the organization's profit, but also improve staff satisfaction with allotted schedule. Depending on the properties of the tasks, there may be a number of constraints due to various scheduling policies. Hence, staff scheduling is formulated as a constrained optimisation problem, which has been shown to be NP-hard [3].

Nurse rostering, which is one type of staff scheduling, intends to automatically allot working shifts to available nurses in order to maximize hospital value/benefit subject to scheduling constraints, including governmental regulations, skill requirement, minimal on-duty hours, nurses' requests, etc. There have been many methods proposed in the last decade for dealing with the nurse rostering problem. These methods can be divided into three categories: (1) mathematical programming techniques (such as branch-and-bound integer programming and linear programming), (2) ad-hoc heuristics, and (3) metaheuristic-based algorithms. Early works [4-7] employing mathematical programming techniques are able to derive exact solutions. However, they are not computationally efficient for solving real-world cases that involve a large number of variables and constraints. Ad-hoc heuristics [8-10] are tailored to the satisfaction of constraints raised in specific nurse-rostering applications and usually employ greedy procedures to improve solutions. The shortcomings of using ad-hoc heuristics are that the greedy search procedures are likely to be trapped by local optima, and customized heuristics are not easily modified to adapt to other problem instances. Metaheuristic algorithms are applicable to a broad class of problems and are a prevalent method in recent literature. Researchers have applied genetic algorithms [11,12], simulated annealing [13], and tabu search [14,15] to obtain nurse schedules in hospitals. An emerging research trend in metaheuristic computing is to enhance swarm algorithms by incorporating the intelligent strategies used in Scatter Search and Path Relinking domains. There are a number of successful applications [16]. Among others, the Cyber Swarm Algorithm [17,18] has been shown to outperform the standard particle swarm optimization algorithm and it is exploited in this study for the application of multiobjective nurse rostering.

Although most of the nurse-rostering literature aimed at a single-objective formulation of the problem, the context in real hospital scenarios shows that the task for nurse rostering involves multiple objectives from variable perspectives. Some commonly seen objectives are minimization of total wage cost, minimal deviation between the nurse demand and the number of nurses actually allotted, maximal satisfaction of nurses' off-duty request, and others. A number of studies [5,7,13,19-22] have attempted to facilitate multiobjective formulation of nurse rostering. Nevertheless, due to high complexity of this multiobjective context, references [5,7,21] converted the multiobjective formulation into a single-objective program by the weighted-sum technique. The weighted-sum technique, however, fails to identify optimal solutions if the Pareto front is non-convex and the value of the weights used to combine multiple objectives is difficult to determine.

This paper proposes a generic formulation of the Multi-Objective Nurse Rostering Problem (MONRP), motivated by the needs of an actual regional hospital. The problem formulation is based on intensive consultations with senior practitioners and administrators affiliated with this hospital. A multiobjective version for the Cyber Swarm Algorithm (CSA) has been developed. The experimental results have shown that the proposed method outperforms state-of-the-art multiobjective evolutionary algorithms on benchmark test functions and real MONRP problems.

The remainder of this paper is organized as follows. Section 2 presents a literature review of existing methods for the nurse rostering problem and introduces the central concepts of multiobjective optimization. Section 3 describes the problem formally using 0-1 linear programming model. Section 4 articulates the proposed method. Section 5 presents simulation results and an analysis of their implications. Finally, concluding remarks and discussions are given in Section 6.

2. **Literature Review.**

This section first presents a comparative review of existing methods for the nurse rostering problem. Then the central ideas of multiobjective optimization are introduced.

2.1. **Nurse rostering problem.**

To assist various operations performed in a hospital 7 days a week and 24h a day, a work day is normally divided into two to four shifts (for example, a three-shift day may include *day*, *night*, and *late* shift). Each nurse is then allocated a number of shifts during the scheduling period with a set of constraints. A shift is fulfilled by a specified number of nurses with different medical skills, depending on the operations to be performed during the shift. The schedule of a nurse can be produced on a shift-by-shift basis as long as the constraints are met, or on a pattern-by-pattern basis, where a pattern is a pre-determined assignment of a set of shifts such that the constraints are easier to handle (such as the maximal number of late shifts). The nurse rostering is usually performed regularly, for instance, all nurse schedules are produced at the same time every other week. But sometimes the nurse schedule is produced dynamically, which happens when a nurse occasionally cannot fulfil the remainder of his/her shifts and has to exchange some shifts with those of others.

The adherent constraints with nurse rostering are necessary hospital regulations when taking into account the wage cost, execution of skilled operations, quality of service, etc. The constraints are classified as hard constraints and soft constraints. Hard constraints must be strictly satisfied and a schedule violating any hard constraints is not accepted. Soft constraints should be satisfied as much as possible and a schedule violating soft constraints is still considered feasible. Typical constraints considered in the literature are summarized as follows and listed in Table 1.

- Constraint (1): The number of shifts allotted to a nurse in a full schedule should be no less than a minimal threshold.
- Constraint (2): The number of nurses with a particular skill scheduled to a given shift should be no less than a minimal threshold.
- Constraint (3): The number of nurses with a particular skill scheduled to a given shift should be no greater than a maximal threshold.
- Constraint (4): The number of consecutive working days allotted to a nurse should be no greater than a maximal threshold.
- Constraint (5): The number of on-duty weekends assigned to a nurse in a full schedule should be no greater than a maximal threshold.
- Constraint (6): The number of a certain shift type assigned to a nurse in a full schedule should be no greater than a maximal threshold.

It can be observed in Table 1 that the most frequently considered constraint in the literature is Constraint (2) which is mandatory to the success of daily operations in hospitals. Constraints (4) and (5) are the second most frequently considered constraints, taking into account the physical workload and weekends off fairness from the nurse's point of view. From the publication year, it can be observed that more types of constraints are

Table 1. Typical constraints considered in literature

Reference, year \ Constraint	(1)	(2)	(3)	(4)	(5)	(6)
[5], 1996	•	•		•	•	
[23], 2000		•				
[14], 2001				•		•
[11], 2003		•				
[15], 2003		•				
[24], 2003	•	•	•	•		•
[25], 2003		•				
[7], 2005	•	•		•	•	
[6], 2005		•	•			
[26], 2007	•	•				
[12], 2007	•	•		•		
[10], 2008		•	•	•	•	•
[19], 2008		•		•	•	
[20], 2009				•		•
[27], 2010		•		•	•	
[28], 2010	•	•		•	•	•
[21], 2010		•	•	•	•	
[13], 2010		•	•	•	•	•
This study	•	•	•	•		•

considered simultaneously in recent papers than those considered in the papers published before 2008. This study extends this research course to include all types of constraints but leaves Constraint (5) as an additional objective to maximize the nurses' preference about the schedule, as will be noted.

The objective of a nurse rostering mainly considers benefits that could be obtained by hospitals or individual nurses. This could involve the reduction of the human resource cost and the satisfaction of nurses' requests. The objectives proposed in the literature are summarized as follows and listed in Table 2.

- Objective (1): Minimization of total wage cost for all nurses.
- Objective (2): Minimization of the total deviation between the minimal number of required nurses and the number of nurses actually allotted.
- Objective (3): Minimization of preference cost for all nurses.
- Objective (4): Minimization of violations to any soft constraints.

Table 2 discloses the fact that the achievement of Objective (3) is the prevailing benefit for hospital administration. The satisfaction of nurse's requests improves the quality of service and maintains good relationship with patients. Another popular formulation of the problem is to transform the soft constraints into Objective (4) by introducing slack/surplus variables and minimizing the variable values, so the obtained solutions will satisfy the soft constraints as much as possible. Specific objectives such as Objective (1) and Objective (2) were considered in [6,7]. These methods keep the number of allotted nurses as small as possible, not only possibly reducing the wage cost but also reserving the flexibility to respond to unexpected situations such as calling back off-duty nurses to substitute for absent ones or to assist incidental missions.

Most existing studies seek to optimize only one of the objectives, and only few consider multiple objectives simultaneously when searching for solutions. Reference [5] presented

Table 2. Typical objectives considered in literature

Objective / Reference, year	(1)	(2)	(3)	(4)
[5], 1996				•
[23], 2000			•	
[14], 2001			•	•
[11], 2003			•	
[15], 2003			•	
[24], 2003			•	
[25], 2003				•
[7], 2005		•		
[6], 2005	•			
[26], 2007			•	
[10], 2008				•
[19], 2008			•	•
[20], 2009				•
[27], 2010			•	
[28], 2010				•
[21], 2010a				•
[13], 2010b			•	•
This study	•	•	•	

the first attempt to find a nurse schedule satisfying several soft constraints simultaneously. The lexico-dominance technique is applied where the priority order of the soft constraints is pre-specified and is used to determine the solution ranking. Reference [14] employed the weighted-sum technique where each objective is assigned a weight value and all objectives are combined by a linear sum of the weighted objectives. Reference [21] also applied the weighted-sum technique, though the weight values were determined by the priority order of objectives obtained after close consultation with hospitals. Instead of using the priority order or transforming multiple objectives into a single one, reference [19] proposed a simple evolutionary algorithm with self-adaptation and population re-generation techniques. The self-adaptation decoder aims to guide the construction of schedules, taking into account hard constraints. The population re-generation strategy rebuilds the population to contain high-quality offspring when the evolutionary process stagnates. Reference [20] used a Multi-Objective Genetic Algorithm (MOGA) to produce non-dominated solutions along generations and store them in an external memory. Reference [13] proposed a simulated annealing multiobjective method which generates the non-dominated solutions to obtain an approximate Pareto front. This study proposes a generic multiobjective formulation of the nurse rostering problem taking into account objective types (1)-(3) simultaneously as shown in Table 2 in order to accommodate complex situations in real hospitals.

2.2. **Multiobjective optimization.** Many real-world problems involve multiple objectives that are conflicting with one another. Typical examples are cost-benefit analysis, engine performance and fuel consumption, weight of a mechanical part and its strength, to name a few. A widely accepted notion for multiobjective optimization is to search for the Pareto-optimal solutions which are not dominated by any other solution. A solution x *dominates* another solution y if x is strictly better than y in at least one objective and x is no worse than y in the others. Formally, given k minimization objective functions, $f_i(x)$, $i = 1, 2, \ldots, k$, solution x dominates solution y, denoted as $x \succ y$, if $f_i(x) \leq f_i(y)$, $\forall i = 1,\ 2,\ \ldots,\ k$ and $f_j(x) < f_j(y)$, $\exists j \in \{1, 2, \ldots, k\}$. The plots of objective values for all Pareto-optimal solutions form a Pareto front in the objective space.

Decision makers are clearly comfortable in seeking Pareto-optimal solutions because if the final solution is not Pareto-optimal it can be improved in at least one objective without deteriorating the solution quality in other objectives. However, it is sometimes hard to find the true Pareto-optimal solutions due to the high complexity of the problem nature, and an approximate Pareto front is instead searched for. The quality of this front is measured in two aspects: (1) the *convergence* metric indicates how closely the approximate Pareto front is converging to the true Pareto front, and (2) the *diversity* metric favors the approximate Pareto front whose plots are most evenly spread. There also exist multiobjective programming approaches such as weighted-sum [29], goal programming [30] and epsilon programming [31]. However, all of these approaches decompose the multiobjective problem into a number of single-objective sub-problems, and multiple executions of the program are needed to obtain the final approximate Pareto front.

Recently, Multi-Objective Evolutionary Algorithms (MOEA) have been introduced as a viable technique to tackle multiobjective optimization problems. Two notable techniques, solution ranking and density estimation, were introduced to obtain high-quality convergence and diversity scores. The solution ranking technique gives each solution found a score and is thus able to perform survival of the fittest to reach good convergence based on the rank of these solutions. The Strength Pareto Evolutionary Algorithm (SPEA2) [32] defines the score for the Pareto strength of a solution by counting the number of other stored solutions dominated by it. The Non-dominated Sorting Genetic Algorithm (NSGA-II) [33] uses the non-dominated sorting strategy which gives the highest score to the solutions non-dominated by the other solutions in the population. The solutions with the highest score are then removed from the population, and the strategy proceeds to give the second-highest score to the non-dominated solutions in the population. The process is repeated until every solution in the population has been assigned a score.

The density estimation technique measures the degree of crowding between the points found in the objective space in order to guide the evolution with good diversity control. SPEA2 uses the k-distance, which is the distance to the k-th nearest neighbour, to estimate the density. The k-distance is more reliable than the shortest distance, which is easily biased by uneven point distributions. As defined in NSGA-II, the crowding distance for a point is the average distance of two points on either side of this point along each of the objectives. The crowding distance is a compromise between the k-distance and the shortest distance and does not require any density parameters. The grid-based technique has been used in the Pareto Archived Evolution Strategy (PAES) [34] and the Multi-Objective Particle Swarm Optimization (MOPSO) [35]. The objective space is divided into regions called grids, and the number of points located in each grid cell is used as an estimate for the density. Another interesting notion for diversity control is via objective decomposition. For example, the MOEA/D algorithm [36] decomposes a multiobjective optimization problem into a number of sub-problems and optimizes them simultaneously. Each sub-problem is defined by a weighted-sum of objectives and the weight vector is well separated from the weight vectors for other sub-problems. Thus, MOEA/D can generate a set of evenly distributed points on the front.

This study proposes a Multi-Objective Cyber Swarm Algorithm (MOCSA) which employs unique strategies as noted in Section 4 to generate a set of points with high quality convergence and diversity performance. The experimental results demonstrate that MOCSA outperforms or performs similarly to main-stream multiobjective evolutionary algorithms in the literature.

3. **Problem Definition.** This paper deals with the Multi-Objective Nurse Rostering Problem (MONRP) on a shift-by-shift basis. Each working day is divided to three shifts (*day*, *night*, and *late* shift), and the shifts in a full scheduling period are numbered from 1 to S (1 indicates the day-shift of the first day, 2 indicates the night-shift of the first day, etc). Assume that there are M types of nurse skills, and skill type m is owned by T_m nurses. The aim of the MONRP is to optimize multiple objectives simultaneously by allotting appropriate nurses to the shifts as long as a set of hard constraints are satisfied. The mathematical formulation of the addressed MONRP is presented as follows.

$$\text{Minimize} \quad f_1 = \sum_{m=1}^{M} \sum_{i=1}^{T_m} \sum_{j=1}^{S} x_{mij} C_{mj} \tag{1}$$

$$\text{Minimize} \quad f_2 = \sum_{m=1}^{M} \sum_{j=1}^{S} \left(\sum_{i=1}^{T_m} x_{mij} - L_{mj} \right) \tag{2}$$

$$\text{Minimize} \quad f_3 = \sum_{m=1}^{M} \sum_{i=1}^{T_m} \sum_{j=1}^{S} x_{mij} \left(1 - P_{mij}\right) \tag{3}$$

$$\text{Subject to} \quad \sum_{j=1}^{S} x_{mij} \geq W_m \quad \forall m, i \tag{4}$$

$$\sum_{i=1}^{T_m} x_{mij} \geq L_{mj} \quad \forall m, j \tag{5}$$

$$\sum_{i=1}^{T_m} x_{mij} \leq U_{mj} \quad \forall m, j \tag{6}$$

$$\sum_{j=r}^{r+2} x_{mij} \leq 1 \quad r = 1, 4, 7, \ldots, S-2 \quad \forall m, i \tag{7}$$

$$\sum_{j=r}^{r+3(R_m+1)-1} x_{mij} \leq R_m \quad r = 1, 4, 7, \ldots, S-2 \quad \forall m, i \tag{8}$$

$$x_{mij} \in \{0, 1\} \quad \forall m, i, j \tag{9}$$

The decision variable x_{mij} takes a binary value and $x_{mij} = 1$ if nurse i having skill m is allotted to shift j; otherwise, $x_{mij} = 0$. The first objective (Equation (1)) intends to minimize the wage cost incurred by performing the nurse schedule as planned where C_{mj} indicates the cost incurred by allotting a nurse having skill m to shift j. The second objective (Equation (2)) tries to minimize the deviation between the minimum number of required nurses for a shift and the number of nurses actually allotted to that shift, L_{mj} presents the minimum number of nurses having skill m required to fulfil the operations in shift j. The third objective initially intends to maximize the total nurses' preference P_{mij} about the planned schedule ($P_{mij} = 1$ if nurse i having skill m is satisfied with shift j assignment; $P_{mij} = -1$ if unsatisfied; and $P_{mij} = 0$ if there is no special preference), it can be converted to a minimization objective by using $1 - P_{mij}$, as shown in Equation (3). Moreover, there are five constraints to be enforced in the planned schedule. The first constraint (Equation (4)) stipulates that the number of shifts allotted to a nurse having skill m should be greater than or equal to a minimum threshold W_m. Equations (5) and (6) describe the next two constraints that the number of nurses having skill m which are allotted to shift j should be a value between the lower bound (L_{mj}) and the upper bound (U_{mj}) as requested. The fourth constraint (Equation (7)) indicates the regulation that any nurse can only work for at most one shift during any working day. Finally, the fifth constraint (Equation (8)) requests that the nurse having skill m can serve for at most R_m consecutive working days. Equation (9) specifies the set of allowable values for the decision variable.

4. **Proposed Method.** The particle swarm optimization (PSO) was first coined by Kennedy and Eberhart [37], and had inspired developments of many extended versions. One of the notable variants is the Cyber Swarm Algorithm (CSA) proposed by [17] which facilitates the reference set to define the acceptance function by retaining a small common memory to store the most influential solutions (by reference to objective value and diversity) from competitions among elite solutions. It is shown in literature that the performance of PSO strongly depends upon the form that it formulates the neighbourhood topology and conducts the swarm intercommunication. The CSA, by exploiting various forms of guiding information and memory programming, has displayed more malleable and effective properties than several existing PSO variants. However, the extension of CSA to solve multiobjective optimization problems is not investigated.

To seek the approximate Pareto optimal solutions for the MONRP problem, this paper proposes the multiobjective version of the CSA, named MOCSA, which adds new

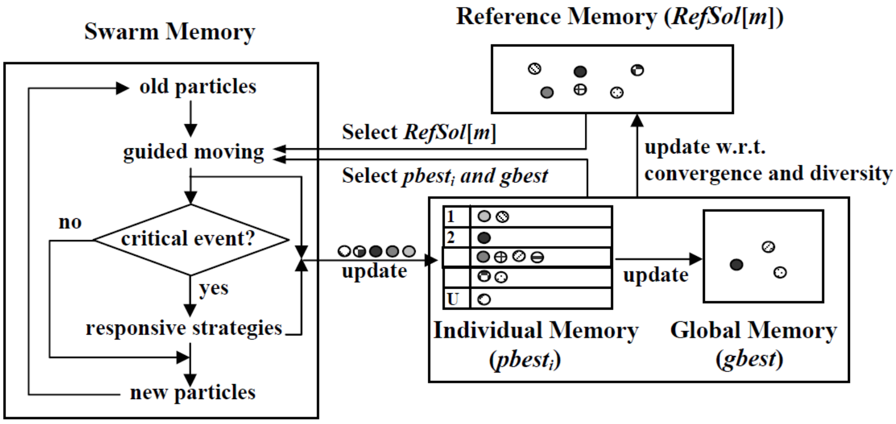

Figure 1. The conception diagram of the MOCSA

features to CSA for generating non-dominated solutions with good convergence and diversity measures. Figure 1 shows the conception diagram of the MOCSA, which consists of four memory components and responsive strategies. The *swarm memory* component is the working memory where the population of swarm particles evolve to improve their solution quality based on guided moving by reference to strategically selected solution guides. The *individual memory* reserves a separate space for each particle and stores the pseudo non-dominated solutions by reference to all the solutions found by the designated particle only. Note that the pseudo non-dominated solutions could be dominated by the solutions found by other particles, but we propose to store the pseudo non-dominated solutions because our preliminary results show that these solutions contain important diversity information along the individual search trajectory and they assist in finding influential solution guides that are otherwise overlooked by just using global non-dominated solutions. The *global memory* tallies the non-dominated solutions that are not dominated by any other solutions found by all the particles. The solutions stored in the global memory will be output as the approximate Pareto-optimal solutions as the program terminates. Finally, the *reference memory* selects the most influential solutions based on convergence and diversity measures. Moreover, to arouse the power of MOCSA when the search loses its efficacy, two responsive strategies are performed upon the detection of critical events which disclose the stagnation of the search power. It should be noted that the manipulation of memory in MOCSA is very different from that used in the original CSA. MOCSA determines the ranking of solutions by the dominance power and diversity relationship in the multi-objective space while the CSA considers the single-objective fitness and the diversity in the solution space. The selection of solution guides is also different in the two versions. In CSA, the best solution leader can be uniquely identified due to the single-objective context. In MOCSA, however, there exist multiple non-dominated solutions in each level of memory and alternative strategies may be applied, as will be noted. Details of the new features of MOCSA are presented in the following subsections.

4.1. **Particle representation and fitness evaluation.** A candidate solution to the MONRP problem consists of schedules for all the nurses in the planning period. Given S working shifts to be fulfilled, there are at most 2^S possible allocations (without considering scheduling constraints) for assigning a nurse to available shifts. Hence, a nurse schedule can be encoded as an integer value between $[0, 2^S - 1]$. Assume that a population of U particles is used in the MOCSA algorithm, and each particle is represented as a vector

$P_i = \{p_{ij}\}$, $0 \le i < U$ and $0 \le j < \sum_{m=1}^{M} T_m$, where $p_{ij} \in [0, 2^S - 1]$ indicates the code of the planned schedule for the jth nurse. The initial population is generated at random.

The fitness of the ith particle is a three-value vector $(f_1,\ f_2,\ f_3)$ whose values are determined by the problem objectives. The objective values evaluated using Equations (1)-(3) are referred to as the three fitness values $(f_1,\ f_2,\ f_3)$, which represent the convergence degree of the obtained solution. As the MONRP is a constrained optimization problem, a penalty value δ is given to an infeasible solution and δ corresponds to the amount of total violations incurred by any constraints (Equations (4)-(8)), namely

$$\begin{aligned}\delta = &\sum_{m=1}^{M}\sum_{i=1}^{T_m}\max\left\{0, \left(W_m - \sum_{j=1}^{S} x_{mij}\right)\right\}\\ &+ \sum_{m=1}^{M}\sum_{j=1}^{S}\max\left\{0, \left(L_{mj} - \sum_{i=1}^{T_m} x_{mij}\right)\right\}\\ &+ \sum_{m=1}^{M}\sum_{j=1}^{S}\max\left\{0, \left(\sum_{i=1}^{T_m} x_{mij} - U_{mj}\right)\right\}\\ &+ \sum_{m=1}^{M}\sum_{i=1}^{T_m}\sum_{r=1,4,..}^{S-2}\max\left\{0,\ \left(\sum_{j=r}^{r+2} x_{mij} - 1\right)\right\}\\ &+ \sum_{m=1}^{M}\sum_{i=1}^{T_m}\sum_{r=1,4,..}^{S-2}\max\left\{0, \left(\sum_{j=r}^{r+3(R_m+1)-1} x_{mij} - R_m\right)\right\}\end{aligned} \tag{10}$$

Hence, the particle corresponds to an infeasible solution if $\delta > 0$. The greater the value of δ, the further the particle is away from the boundary of the solution feasibility. On the other hand, the particle represents a feasible solution if $\delta = 0$, indicating that all constraints are met. In order to handle solution selection with the presence of both feasible and infeasible solutions, the constrained-dominance relationship used in both NSGA-II and MOPSO is adopted in this paper. A solution x is said to constrained-dominate a solution y, if any of the following conditions is true: (a) solutions x and y are feasible and solution x dominates solution y by reference to the three fitness values (f_1, f_2, f_3), (b) solutions x and y are both infeasible and solution x has a smaller value of δ, and (c) solution x is feasible while solution y is not. For simplicity of presentation, "dominate" indicates "constrained dominate" in the remainder of this paper, unless otherwise stated.

The use of constraint handling strategy can gradually reduce the probability of producing infeasible solutions during the evolution. When two infeasible solutions are present, the infeasible solution with a smaller constraint violation is selected. The constraint handling strategy also helps sort out non-dominated solutions among the feasible solutions.

4.2. **Exploiting guiding information.** The traditional PSO conducts swarm learning by using two solution guides: (1) the personal best experience, denoted *pbest*, which has been recognized in individual historical behaviours; and (2) the global best experience, denoted *gbest*, which has been observed by the entire swarm. The CSA extends the learning form by additionally including another solution guide that is systematically selected from the reference set, a notion from scatter search [38]. The reference set stores a small number of reference solutions, denoted *RefSol*$[m]$, $m = 1, 2, \ldots,$ RS, observed by all particles with reference to fitness values and solution diversity. To implement the MOCSA, the selection of solution guides is more complex because multiple non-dominated solutions can play the role of *pbest*, *gbest* and *RefSol*$[m]$. The updating and selection for solution guides are explained in the following subsections. Once the three solution guides were selected, particle P_i updates its positional vector in the swarm memory by the guided moving using Equations (11) and (12), as follows.

$$v_{ij}^m \leftarrow K\left(v_{ij} + (\varphi_1+\varphi_2+\varphi_3)\left(\frac{\omega_1\varphi_1 pbest_{ij} + \omega_2\varphi_2 gbest_j + \omega_3\varphi_3 RefSol[m]_j}{\omega_1\varphi_1+\omega_2\varphi_2+\omega_3\varphi_3} - p_{ij}\right)\right), \quad 1 \le m \le RS \quad (11)$$

$$P_i \leftarrow \text{constrained-dominance}\left\{P_i + v_i^m \,|m \in [1, RS]\right\} \quad (12)$$

where K is the constriction factor, ω and φ are the weighting value and the cognition coefficient for the three solution guides *pbest*, *gbest* and *RefSol* $[m]$. As *RefSol*$[m]$, $1 \le m \le RS$ is selected in turn from the reference set, the process will generate RS candidate particles for replacing P_i. Then the non-dominated solution(s) among the RS candidate particles according to the constrained-dominance relationship is selected. If there exist more than one non-dominated solutions, an arbitrary one is used to replace P_i. Nevertheless, to expedite collective learning, all the non-dominated solutions found in the guided moving are used for experience memory update as noted in the following.

4.3. **Experience memory update.** As shown in Figure 1, experience memory consists of individual memory, global memory and reference memory, where the candidate solutions for serving as local guides (*pbest*, *gbest* and $RefSol[m]$) are selected. There are a number of selection strategies for selecting local guides from the experience memory, as will be noted in the next subsection. In what follows, the method for experience memory update is presented.

The individual memory tallies the personal rewarded experience for each individual particle. Because there may exist more than one non-dominated solution in the search course of a particle (here, non-dominance only refers to all the solutions found by this particle), all these solutions are stored in the individual memory. In contrast to individual memory, the global memory stores all the non-dominated solutions found by any particles (here, non-dominance refers to all the solutions found by the entire swarm). Hence, the content of the global memory is used for the final output of the approximate Pareto-optimal solutions when the algorithm terminates. The reference memory stores a small number of reference solutions selected from individual and global memory. In the original scatter search template [38] the solutions are separately sorted based on the single-objective value and the density distance in solution space. In the context of multiobjective optimization, however, the convergence and diversity of the finally obtained front in the objective space are of major concern. Consequently, the 2-tier reference memory update proceeds as follows. Let the reference memory contain RS reference solutions. The first tier consists of $RS/2$ good convergence solutions selected from the global memory according to the constrained-dominance ranking. The second tier contains the other $RS/2$ ordered high diversity solutions selected from the individual memory. The solution diversity is measured in the objective space and is ordered as follows. For each solution in the individual memory, the minimum distance (measured in the objective space) to each of the current members in the reference memory is computed. Then the solution with the maximum of these minimum distances is added to the reference memory. This max-min selection process is repeated until the reference memory contains RS solutions. Because the reference memory is updated in consideration of both convergence and diversity in the objective space, the reference solutions are good local guides for leading the particles to approximate Pareto front.

4.4. **Selecting solution guides.** As mentioned in Section 4.2, *pbest*, *gbest* and $RefSol[m]$ are used for the guided moving of a particle. These three solution guides are empirically found to be influential in improving the convergence (minimization of objective values) and diversity (uniform distributions of plots on the front) performance. The selection strategy for each solution guide is described as follows.

Reference [39] studied a number of selection strategies for selecting *pbest*. They showed that keeping a personal memory for storing all the visited non-dominated solutions has better performance than keeping only one value for *pbest* and updating it as needed. Among the alternative strategies, the *Diversity strategy* was empirically shown to outperform the others (such as Random, Weighted Sum of Objective Values, and Average of All *pbest* Candidates) on a set of benchmark test functions. Consequently, the Diversity strategy for selecting *pbest* is employed in our method and is described as follows. The Diversity strategy stipulates that each particle selects from its individual memory a non-dominated solution as *pbest* that is the furthest away from the other particles in the objective space. Thus, the guided particle is likely to produce a plot of objective values equal-distanced to those of other particles, improving the diversity measure of the solution front.

The selection for *gbest* can also be implemented by a number of strategies. Reference [40] proposed the *Sigma strategy*, which has been shown to compare favorably to SPEA2 [32] and D-tree [41]. The Sigma strategy was implemented in MOCSA for selecting *gbest* from the global memory. For a given particle, the Sigma strategy selects from the global memory a non-dominated solution as *gbest*, which is the closest to the line connecting the plot of the particle's objective values to the origin in the objective space. Therefore the Sigma strategy induces the particles to fly toward the Pareto front, improving the convergence measure of the finally produced front.

Finally, the third solution guide, $RefSol[m]$, $m = 1, 2, \ldots, RS$, is selected in turn from the reference memory. These reference solutions have good properties of convergence and diversity as noted in the reference memory update method, so their features are fully explored in the guided moving for a particle.

4.5. **Responsive strategies.** Critical events are indicators disclosing that the search course may have lost its search efficacy. Purposeful strategies responding to critical events observed in longer term search history are helpful in finding solutions having features differing from those previously seen. A commonly used responsive strategy technique is path relinking (PR) [38], which is a search process that constructs a link between two strategically selected solutions. The construction starts with one solution (called the initiating solution) and moves to a second solution (referred to as the guiding solution). PR transforms the initiating solution into the guiding solution by generating moves that successively replace an attribute of the initiating solution by the attribute that is contained in the guiding solution. The best fitness solution identified along the constructed link is used to restart a new search course.

Two responsive strategies are proposed in MOCSA for the purpose of improving the convergence and diversity of the produced front, as follows. (1) The critical event that the global memory has not been updated for t_1 iterations indicates the whole swarm has lost its search efficacy. Thus, the Convergence PR strategy is activated to restart every particle in the swarm by exploiting the region between its two closest *gbest* in the objective space. (2) The Diversity PR strategy is performed upon the critical event that a particular particle's individual memory has not been updated for t_2 iterations, disclosing that this particle has been trapped by a local optimum. The strategy reinitiates this particle by

1 Initialization

1.1 Randomly generate U particle solutions, $P_i = \{p_{ij}\}$, $0 \le i < U$, $0 \le j < \sum_{m=1}^{M} T_m$

1.2 Randomly generate U velocity vectors, $V_i = \{v_{ij}\}$, $1 \le i \le U$, $0 \le j < \sum_{m=1}^{M} T_m$

1.3 Evaluate multiple fitness values for each particle. Update experience (individual, global, and reference) memory

2 Repeat until a stopping criterion is met

2.1 For each particle P_i, $\forall i = 1, \ldots, U$, Do

2.1.1 **Guided moving with selected solution guides:**

$$v_{ij}^m \leftarrow K\left(v_{ij} + (\varphi_1 + \varphi_2 + \varphi_3)\left(\frac{\omega_1\varphi_1 pbest_{ij} + \omega_2\varphi_2 gbest_{ij} + \omega_3\varphi_3 RefSol[m]_j}{\omega_1\varphi_1 + \omega_2\varphi_2 + \omega_3\varphi_3} - p_{ij}\right)\right)$$

$P_i \leftarrow$ constrained-dominance $\left\{ P_i + v_i^m \;\middle|\; m \in [1, RS] \right\}$

2.1.2 Update experience memory if necessary

2.2 **Convergence PR strategy:** If the global memory has not been updated for t_1 iterations, restart every particle by exploiting the region between the particle's two closest *gbest* (say $gbest_j$ and $gbest_k$) by

$P_i \leftarrow$ Convergence_PR($gbest_j$, $gbest_k$), $\forall i = 1, \ldots, U$

Diversity PR strategy: Else if a particular particle's individual memory has not been updated for t_2 iterations, restart this particle by

$P_i \leftarrow$ Diversity_PR($pbest_i$, P_i)

2.3 Update experience memory if necessary

3 Output feasible non-dominated solutions in the global memory

FIGURE 2. Pseudo codes for the multiobjective CSA (MOCSA)

performing PR between the particle itself and its *pbest* (using the *pbest* selection method noted in Section 4.4).

4.6. **Algorithm overview and advantages.** In summary, the proposed MOCSA is outlined in Figure 2. The initialization (Step 1) gives the initial values for particle positions, velocities, and experience memory. The evolutionary iterations (Step 2) perform the guided moving, memory update, and responsive strategies. In Step 2.1, each particle moves with the guidance information provided by three carefully selected solution guides ($pbest_i$, *gbest* and $RefSol[m]$, $m = 1, \ldots, RS$). The guided moving is repeatedly conducted and each instance uses a different member from the reference memory. All the constrained non-dominated solutions identified in the multiple movement trials are used for experience memory update. In Step 2.2, responsive PR strategies are employed upon critical events for improving the convergence and diversity of the currently produced front. When the stopping criterion of MOCSA is met, all the feasible non-dominated solutions in the global memory are used to plot the approximate Pareto front.

The advantages of the MOCSA algorithm compared with previous MOEA algorithms for improving convergence and diversity performances are as follows. (1) MOCSA adopts multi-level memory to facilitate solution ranking and density estimation mechanisms. The experience update is conducted in the order of swarm, individual, global and reference memory, and the best experience obtained in a low level memory is used for update of the next level memory. Thus the multi-level memory realizes the solution ranking in guiding the evolution. Moreover, as each type of memory performs a different update strategy which will preserve unique features of its maintained solutions, sustaining good diversity in the whole population. (2) The reference set is an adaptive memory that stores the most influential solutions by reference to convergence and diversity estimates.

This feature is very effective in guiding the revolution towards the true Pareto front. (3) The solution guides are selected according to competitions related to both performance measures. These solution guides are used systematically in combination with a reference solution selected in turn from the reference memory, imposing a dynamic social network for fostering a new particle. (4) Two responsive strategies triggered by critical events are particularly developed for improving the convergence and diversity performances since MOEA algorithms usually suffer from premature convergence. The responsive strategies employed in MOCSA are useful in detecting these critical events and redirecting the search towards uncharted regions.

5. **Result and Discussion.** In order to testify the robustness and effectiveness of the proposed MOCSA, both of the benchmark test functions and the MONRP instances were experimented with for performance comparison with the state-of-the-art MOEA algorithms. All the algorithms were coded using C# language, and the following experiments were conducted on a 2.4GHz PC with 1.25GB RAM.

5.1. **Test datasets.** Two test datasets were used in our experiments and they are described as follows.

- Benchmark ZDT dataset. This dataset is the benchmark multiobjective test functions [42] which have been intensively used in the literature. All the functions are unconstrained optimization problems and have two objectives. The ZDT functions have Pareto fronts of various natures and the Pareto-optimal solutions are known, making them suitable for comparing the convergence and diversity performances of various multiobjective optimization methods.
- Simulation MONRP dataset. This dataset was created by intensively consulting the administrators and senior staffs at the Puli Christian Hospital (http://www.pch.org.tw/english/e_index.html). The MONRP dataset consisting of two problem instances was used to assess the objective values of the nurse schedules produced by various algorithms. According to consultation with the administrators and senior staff at this hospital, a working day contains three shifts (*day*, *night*, and *late*) and the nurse schedule is determined on a shift-by-shift basis. The first problem instance (Problem I) requires that the optimal scheduling of 10 nurses with two levels of skills in a planning period of one week is determined, while the second problem instance (Problem II) consists of 25 nurses with three different skills to be scheduled in a period of four weeks.

5.2. **Competing algorithms.** We have chosen the state-of-the-art MOEA algorithms for performance comparison because they have been tested on the benchmark ZDT dataset and the results are available in the literature. These algorithms are NSGA-II [33], MOPSO [35], MOEA/D [36], SPEA [32] and PAES [34]. The NSGA-II and MOPSO have also been implemented in this study for comparison with the proposed MOCSA algorithm on the simulation MONRP dataset.

5.3. **Performance measures.** As previously noted, the quality of the solution front found by each algorithm should be evaluated in two types of metrics, the *convergence* and the *diversity*. A number of notable performance indexes broadly used in the literature are adopted in our experiment and they are described as follows.

- Number of points (n). This refers to the number of obtained solutions, and it also is the number of obtained points in the objective space. Since decision makers prefer referring to a greater number of non-dominated solutions, the larger the n value the better.

- Generational distance (GD). Reference [43] proposed the GD metric, which estimates how far the points generated by a given algorithm are from the true Pareto front. Assuming that there are n points generated in the objective space, for each of the points, let Δ_i be the distance between this point and its closest efficient point on the Pareto front. GD is derived by $\mathrm{GD} = \sqrt{\sum_{i=1}^{n} \Delta_i^2} \Big/ n$. The smaller the GD value, the better the obtained solutions.
- Error ratio (ER). ER corresponds to the ratio accounting for the percentage of the generated solutions which are not Pareto optimal. Let e_i be a binary variable that is equal to zero if the ith generated solution is a member of the true Pareto optimal set; otherwise, it is equal to one. ER can be derived using $ER = \sum_{i=1}^{n} e_i \Big/ n$. The smaller the ER value, the more likely the generated solutions are to be Pareto optimal.
- Size of the space covered (SSC). Reference [44] proposed this metric for measuring the volume of points dominated by the produced front. In particular, the SSC sums up the volume enclosed by the produced front and a reference point dominated by all the obtained solutions on the front. High values of SSC indicate good quality in both of convergence and diversity of the obtained solutions. Hence, the larger the SSC value, the better the produced front.
- Spacing (SP). Reference [45] proposed a diversity measure named Spacing which estimates the spread of the obtained points in the objective space. Let d_i be the distance between the ith point and its closest point, and $\overline{d}$ the mean of all d_i. The SP metric is defined as $\sqrt{\frac{1}{n-1} \sum_{i=1}^{n} \left(\overline{d} - d_i\right)^2}$. The produced solutions with a smaller SP value are more desirable because these solutions exhibit better representation of a spread front. The minimum value of SP is zero, which indicates that the points of all the generated solutions are equal-distanced in the objective space.
- Density estimation (k-distance). This metric was used by [32] in implementing the density estimation method for SPEA2. The k-distance of each generated point refers to the distance to the kth nearest point in the objective space. The mean and the maximum of k-distance values reveal the point density information. A smaller k-distance value denotes a better approximation in terms of frontier density. In our analysis, k is equal to the square root of the sample size as suggested in [32].
- Coverage of two sets (C(A, B)). Reference [44] proposed this performance metric for comparison of two competing multiobjective algorithms. Let A and B be the sets of solutions produced by two algorithms. The ordered measure C(A, B) calculates the percentage of members of B that are dominated by any members of A. Thus, both C(A, B) and C(B, A) must be considered.

5.4. **Simulation results.** As all of the competing algorithms are stochastic, meaning that each run of a given algorithm on the same problem instance may return a different result. The mean value of performance measures over 30 independent runs is reported. Two experiments were conducted in this study. Experiment I compares the proposed MOCSA algorithm with the state-of-the-art MOEA algorithms on the benchmark ZDT test functions and uses the results to justify the contribution of various strategies for solution guide selection used in MOCSA. Experiment II evaluates the performance of MOCSA for the MONRP problem. Two of the distinguished methods, NSGA-II and MOPSO, indicated in Experiment I are used for comparison. The parameter setting of the implemented algorithms is tabulated in Table 3. These parameter values are determined according to the design of experiments (DOE) principle. Typical values of each parameter have been tried and the ones giving the best performance are retained. The crossover rate and mutation rate for NSGA-II and MOPSO are set to the values used in the original papers. The NSGA-II was executed with binary coding using uniform crossover and flipping mutation.

A. Experiment I: Benchmark Multiobjective Test Functions

The ZDT test functions were designed to simulate various scenarios of multiobjective problems such as multimodality, high dimensionality, and discontinuity. These are useful in analyzing properties of multiobjective optimization methods because the true Pareto-optimal solutions to these functions are known. Many MOEA algorithms have been tested on the ZDT test functions and most of them have reported the GD and SP performance values in the original papers. Tables 4 and 5 list the computational results obtained by these algorithms for GD and SP metrics, respectively. It can be seen that MOCSA obtains the best result for all functions in terms of both metrics except for the GD value on ZDT4 function. MOPSO obtains the second-best value for most of the functions and NSGA-II performs particularly well on ZDT4 function. Except for the three notable algorithms, MOEA/D performs better than the remaining algorithms by reference to the GD value (the SP value for MOEA/D was not reported in the original paper). SPEA and PAES seem to perform worse than the other competitors.

NSGA-II and MOPSO are among the best algorithms observed in Tables 4 and 5, and so they are implemented in this study for justifying the performance of alternative strategies

Table 3. The setting of parameter values for implemented algorithms

MOCSA	Swarm size	Reference memory size	Individual memory size	Global memory size	t_1	t_2
	100	10	1000	200	6	10
NSGA-II	Population size	Crossover rate	Mutation rate			
	200	0.8	$1/string\_length$			
MOPSO	Swarm size	Archive size	Mutation rate			
	100	200	0.5			

Table 4. The GD values obtained by various algorithms

GD metric	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
MOCSA	0.00034	0.00035	0.00302	8.45617	0.00041
NSGA-II	0.000894	0.000824	0.043411	3.227636	7.806798
MOPSO	0.000428	0.000421	0.003307	35.690697	0.016530
MOEA/D	0.0055	0.0079	0.0143	0.0076	0.0042
SPEA	0.001799	0.001339	0.047517	7.340299	0.221138
PAES	0.082085	0.126276	0.023872	0.854816	0.085469

Table 5. The SP values obtained by various algorithms

SP metric	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6
MOCSA	0.002012	0.001466	0.002573	0.442721	0.020167
NSGA-II	0.463292	0.435112	0.575606	0.479475	0.644477
MOPSO	0.003543	0.003888	0.004752	0.956675	0.035805
SPEA	0.784525	0.755148	0.672938	0.798463	0.849389
PAES	1.229794	1.165942	0.789920	0.870458	1.153052

for selecting solution guides in MOCSA, as noted in Section 4.4. Various coupled settings for these strategies have been used. For *pbest* selection, both the Random and Diversity selection strategies are tested. For *gbest* selection, the Random, Sum (of objective values) and Sigma selection strategies are used, respectively. For each ZDT function, the rank on each of the four performance measures, GD, ER, SSC, and SP, are derived, and the sum of ranks are used to determine the final rank of the ZDT function. Similarly, the overall rank of individual strategies is derived from the sum of ranks over all five ZDT functions. The rank information is used because this would avoid the scaling effect and distribution discrepancy of original values for different performance measures.

It can be seen in Table 6 that the MOCSA algorithm coupled with Diversity *pbest* selection strategy constantly performs better than that with Random *pbest* selection strategy for all ZDT functions in terms of the overall rank. This shows the effectiveness of Diversity strategy in improving the point diversity on the produced front. For the MOCSA coupled with various *gbest* selection strategies, Sigma strategy is better than the other two counterparts. In comparison with NSGA-II and MOPSO, the Diversity strategy coupled with any *gbest* selection strategies can obtain the best overall rank. It should also be noted that NSGA-II gets the first rank among all the competitors for solving ZDT3 and ZDT4, but it gets the worst rank for the rest of ZDT functions. The MOPSO ranks at the middle for solving ZDT1 and ZDT6, and its rank varies significantly for the other ZDT functions.

B. Experiment II: Multiobjective Nurse Rostering Problems

The best form of MOCSA, Diversity with Sigma, identified in Experiment I is applied to solve the MONRP problem and it is compared to NSGA-II and MOPSO. Since deriving the true Pareto front to the simulation MONRP dataset is computationally prohibitive, the performance measures of GD and ER are not applicable. The values of the rest of the performance indexes are shown in Tables 7-10. Table 7 lists the mean value of N, SSC, k-distance, and SP measures obtained from the 30 independent runs of compared algorithms for MONRP Problem I. The MOCSA algorithm has the best performance in terms of N and SSC values, indicating that MOCSA exhibits good convergence capability by producing more points to represent the front and dominating a larger area. The MOCSA also maintains good diversity among the points by being able to rank at the middle for k-distance and SP metrics. The NSGA-II produces the smallest values for k-distance and SP metrics, showing the best spacing between adjacent points. The overall rank of each algorithm is determined by summing the rank for each metric (the average rank for the mean and the maximum k-distance is referred to as one item to avoid overweighting). MOCSA ranks in the first place, NSGA-II obtains the second place, and MOPSO performs worst for Problem I. The experimental outcome for Problem II is shown in Table 8.

Table 6. The performance rank of various strategies for ZDT functions

pbest Selection	*gbest* Selection	ZDT1	ZDT2	ZDT3	ZDT4	ZDT6	Overall Rank
Random	Random	6	7	7	8	7	8
	Sum	3	2	8	6	5	5
	Sigma	6	3	6	4	6	6
Diversity	Random	2	3	2	2	3	2
	Sum	5	5	5	5	1	3
	Sigma	1	1	4	3	1	1
NSGA-II		8	8	1	1	8	7
MOPSO		4	6	2	7	4	4

Table 7. The mean value of performance indexes obtained by competing algorithms for solving Problem I

	N	SSC	k-distance Mean	Max	SP	Overall Rank
MOCSA	14.6	2.83E+06	2622.98	5050.01	1.98	1
NSGA-II	12.1	2.14E+06	2554.82	4000.00	1.62	2
MOPSO	8.8	2.28E+06	3541.79	6350.01	2.24	3

Table 8. The mean value of performance indexes obtained by competing algorithms for solving Problem II

	N	SSC	k-distance Mean	Max	SP	Overall Rank
MOCSA	13.6	7.39E+06	2201.09	4750.01	3.16	1
NSGA-II	13.3	7.18E+06	5205.04	9650.02	4.12	2
MOPSO	11.5	6.91E+06	2389.91	4400.03	5.95	3

Table 9. The values of C(A, B) measure for solving Problem I

A \ B	MOCSA	NSGA-II	MOPSO
MOCSA	–	0.70	0.67
NSGA-II	0.15	–	0.22
MOPSO	0.08	0.30	–

Table 10. The values of C(A, B) measure for solving Problem II

A \ B	MOCSA	NSGA-II	MOPSO
MOCSA	–	0.05	0.60
NSGA-II	0.45	–	0.62
MOPSO	0.00	0.00	–

Again, MOCSA produces the best values for almost all of the metrics, thus obtaining the best overall rank. The second-best overall rank is given to NSGA-II because it performs better than MOPSO in N, SSC, and SP values.

The C(A, B) metric is useful in comparing two competing algorithms. Table 9 shows the values of the C(A, B) metric for solving Problem I by considering A at the head of each row and B at the top of each column. The solutions obtained by MOCSA dominate 70% and 67% of the solutions produced by NSGA-II and MOPSO, respectively. On contrary, only 15% and 8% of MOCSA solutions are dominated by the solutions of NSGA-II and MOPSO. Moreover, MOPSO seems to have stronger dominance power than NSGA-II by producing a larger value (30%) of the C metric than its counterpart (22%). However, MOPSO may produce worse frontier diversity than NSGA-II, as previously revealed in the k-distance and SP values shown in Table 7.

The result of C values for solving Problem II by the test algorithms is shown in Table 10. NSGA-II has the best dominance power and its solutions dominate 45% and 62% of the solutions generated by MOCSA and MOPSO, respectively, while very few of NSGA-II

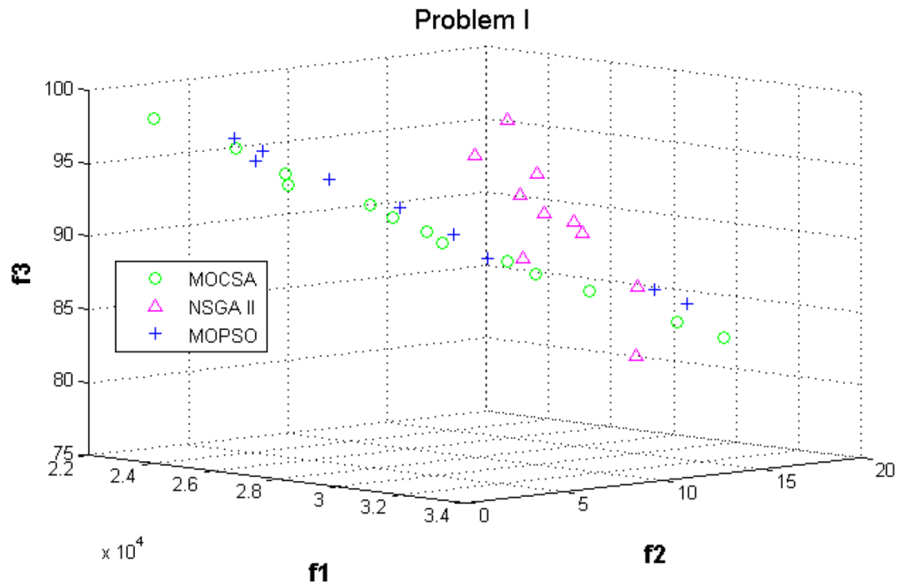

FIGURE 3. The multiobjective-valued front for Problem I

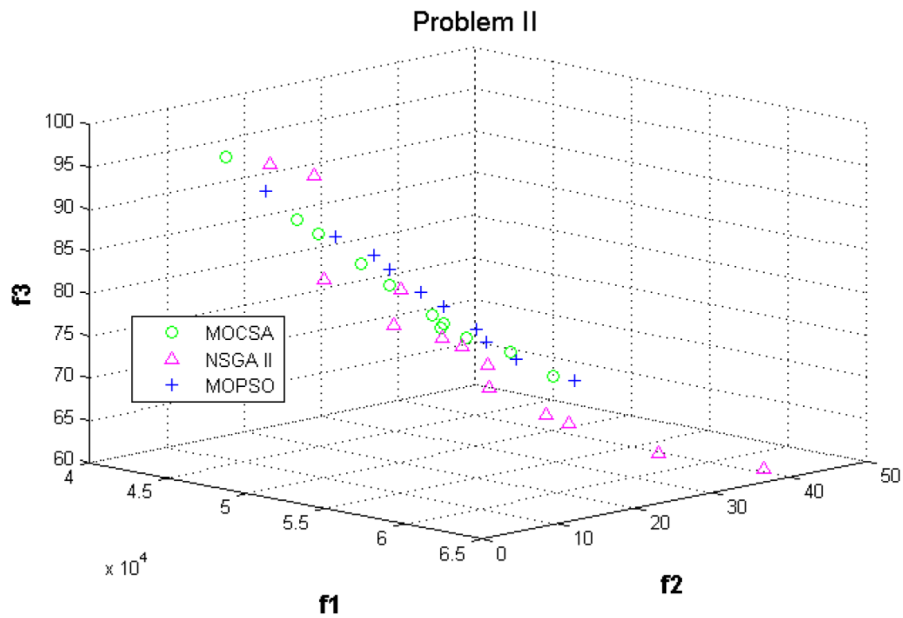

FIGURE 4. The multiobjective-valued front for Problem II

solutions are dominated by its counterparts. MOCSA outperforms MOPSO by dominating 60% of MOPSO solutions and producing no solutions dominated by MOPSO. It is noted that the C metric should be used with other metrics for a careful explanation. Although NSGA-II has the best C values, MOCSA may produce a result with better frontier diversity, as seen in the k-distance and SP values in Table 8, and the SSC value also indicates that MOCSA can dominate a larger space of points.

For a visual comparison of the solution fronts produced by various algorithms, Figure 3 shows the plots of the objective values of the obtained solutions for Problem I by different algorithms. It can be seen that the front produced by MOCSA is closer to the origin, hence obtaining the greatest SSC value, as shown in Table 7. We can also observe that the spread of the solutions obtained by MOCSA are better distributed on the front than those produced by the other two methods. The front generated by MOPSO is next to that produced by MOCSA by reference to the visual distance to the origin. But the front generated by NSGA II does not converge well. For Problem II, the produced fronts by various algorithms are shown in Figure 4. We can see this time that the front produced by NSGA II converges similarly to that obtained by MOCSA, but the latter exhibits better spacing among the produced points. The front produced by MOPSO is furthest from the origin, indicating that MOPSO may not be well scalable to large MONRP problems.

6. **Conclusions.** In this paper, a multiobjective cyber swarm algorithm (MOCSA) for solving the nurse rostering problem has been presented. From our literature survey on nurse rostering, a mathematical formulation that contains three objectives and five hard constraints is proposed. In contrast to most existing methods which transform multiple objective values into a weighted sum, the proposed MOCSA method tackles the generic multiobjective setting and it is able to produce approximate Pareto front. The MOCSA incorporates salient features from particle swarm optimization, adaptive memory programming, scatter search and path relinking to create benefit from synergy. The experimental results on two datasets demonstrate that MOCSA outperforms several MOEA algorithms in terms of convergence and diversity measures of the produced fronts.

Acknowledgment. This research is partially supported by National Science Council of Taiwan, under Grant NSC 98-2410-H-260-018-MY3.

REFERENCES

[1] C. Chiu, M. J.-J. Wu, Y.-T. Tsai, N.-H. Chiu, M. S.-H. Ho and H.-J. Shyu, Constrain-based particle swarm optimization (CBPSO) for call center scheduling, *International Journal of Innovative Computing, Information and Control*, vol.5, no.12(A), pp.4541-4549, 2009.

[2] A. T. Ernst, H. Jiang, M. Krishnamoorthy and D. Sier, Staff scheduling and rostering: A review of applications methods and models, *European Journal of Operational Research*, vol.153, pp.3-27, 2004.

[3] M. J. Brusco and T. R. Johns, A sequential integer programming method for discontinuous labor tour scheduling, *European Journal of Operational Research*, vol.95, pp.537-548, 1996.

[4] M. Warner and J. Prawda, A mathematical programming model for scheduling nursing personnel in a hospital, *Management Science*, vol.19, pp.411-422, 1972.

[5] I. Berrada, J. A. Ferland and P. Michelon, A multi-objective approach to nurse scheduling with both hard and soft constraints, *Socio-Economic Planning Sciences*, vol.30, pp.183-193, 1996.

[6] J. F. Bard and H. W. Purnomo, Preference scheduling for nurses using column generation, *European Journal of Operational Research*, vol.164, pp.510-534, 2005.

[7] M. N. Azaiez and S. S. Al Sharif, A 0-1 goal programming model for nurse scheduling, *Computers & Operations Research*, vol.32, pp.491-507, 2005.

[8] L. D. Smith, D. Bird and A. Wiggins, A computerized system to schedule nurses that recognizes staff preferences, *Hospital & Health Service Administration*, pp.19-35, 1979.

[9] R. Blau, Multishift personnel scheduling with a microcomputer, *Personnel Administrator*, vol.20, pp.43-58, 1985.

[10] E. K. Burke, T. Curtois and G. Post, A hybrid heuristic ordering and variable neighbourhood search for the nurse rostering problem, *European Journal of Operational Research*, vol.188, pp.330-341, 2008.

[11] U. Aickelin and K. A. Dowsland, An indirect genetic algorithm for a nurse scheduling problem, *Computers & Operations Research*, vol.31, pp.761-778, 2003.

[12] M. Moz and M. V. Pato, A genetic algorithm approach to a nurse rerostering problem, *Computers & Operations Research*, vol.34, pp.667-691, 2007.

[13] E. K. Burke, J. Li and R. Qu, A Pareto-based search methodology for multi-objective nurse scheduling, *Annals of Operations Research*, 2010.

[14] F. Bellanti, G. Carello, F. D. Croce and R. Tadei, A tabu search approach to a nurse rostering, *Proc. of the 4th Metaheuristics International Conference*, pp.165-167, 2001.

[15] E. K. Burke, G. Kendall and E. Soubeig, A tabu-search hyperheuristic for timetabling and rostering, *Journal of Heuristics*, vol.9, pp.451-470, 2003.

[16] M. G. H. Omran, Special issue on scatter search and path relinking methods, *International Journal of Swarm Intelligence Research*, vol.2, no.2, 2011.

[17] P. Y. Yin, F. Glover, M. Laguna and J. X. Zhu, Cyber swarm algorithms – Improving particle swarm optimization using adaptive memory strategies, *European Journal of Operational Research*, vol.201, pp.377-389, 2010.

[18] P. Y. Yin and E. P. Su, Cyber Swarm optimization for general keyboard arrangement problem, *International Journal of Industrial Ergonomics*, vol.41, pp.43-52, 2011.

[19] D. Landa-Silva and K. N. Le, A simple evolutionary algorithm with self-adaptation for multi-objective nurse scheduling, adaptive and multilevel metaheuristics, *Studies in Computational Intelligence*, vol.136, pp.133-155, 2008.

[20] O. Sharif, A. Ünveren and A. Acan, Evolutionary multi-objective optimization for nurse scheduling problem, *Proc. of the ICSCCW'2009*, 2009.

[21] E. K. Burke, J. Li and R. Qu, A hybrid model of integer programming and variable neighbourhood search for highly-constrained nurse rostering problems, *European Journal of Operational Research*, vol.203, pp.484-493, 2010.

[22] P. Y. Yin, C. C. Chao and Y. T. Chiang, Multiobjective optimization for nurse scheduling, *Proc. of the International Conference on Swarm Intelligence*, Chongqing, China, pp.66-73, 2011.

[23] A. Jan, M. Yamamoto and A. Ohuchi, Evolutionary algorithms for nurse scheduling problem, *Proc. of the 2000 Congress on Evolutionary Computation*, vol.1, pp.196-203, 2000.

[24] B. Cheang, H. Li, A. Lim and B. Rodrigues, Nurse rostering problems – A bibliographic survey, *European Journal of Operational Research*, vol.151, pp.447-460, 2003.

[25] J. Li and U. Aickelin, A Bayesian optimization algorithm for the nurse scheduling problem, *Proc. of the 2003 Congress on Evolutionary Computation*, 2003.

[26] U. Aickelin, E. K. Burke and J. Li, An estimation of distribution algorithm with intelligent local search for rule-based nurse rostering, *Journal of the Operational Research Society*, vol.58, pp.1574-1585, 2007.

[27] R. Bai, E. K. Burke, G. Kendall, J. Li and B. McCollum, An efficient evolutionary hybrid algorithm for the nurse rostering problem, *IEEE Transactions on Evolutionary Computation*, 2010.

[28] P. Brucker, E. K. Burke, T. Curtois, R. Qu and G. V. Berghe, A shift sequence based approach for nurse scheduling and a new benchmark dataset, *Journal of Heuristics*, vol.16, pp.559-573, 2010.

[29] P. C. Chang, S. H. Chen and V. Mani, Parametric analysis of bi-criterion single machine scheduling with a learning effect, *International Journal of Innovative Computing, Information and Control*, vol.4, no.8, pp.2033-2043, 2008.

[30] M. J. Schniederjans, *Goal Programming: Methodology and Applications*, Kluwer Academic Publishers, Norwell, USA, 1995.

[31] K. Yokoyama, Epsilon approximate solutions for multiobjective programming problems, *Journal of Mathematical Analysis and Applications*, vol.203, pp.142-149, 1996.

[32] E. Zitzler, M. Laumanns and L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm, *Technical Report 103, Swiss Federal Institute of Technology (ETH)*, Switzerland, 2001.

[33] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, vol.6, pp.42-50, 2002.

[34] J. D. Knowles and D. W. Corne, Approximating the nondominated front using the Pareto archived evolution strategy, *Evolutionary Computation*, vol.8, pp.149-172, 2000.

[35] C. A. C. Coello, G. T. Pulido and M. S. Lechuga, Handling multiple objectives with particle swarm optimization, *IEEE Transactions on Evolutionary Computation*, vol.8, pp.256-279, 2004.

[36] Q. Zhang and H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Transactions on Evolutionary Computation*, vol.11, no.6, pp.712-731, 2007.

[37] J. Kennedy and R. C. Eberhart, Particle swarm optimization, *Proc. of IEEE Int'l. Conf. on Neural Networks*, vol.4, pp.1942-1948, 1995.

[38] F. Glover, A template for scatter search and path relinking, artificial evolution, *Lecture Notes in Computer Science*, vol.1363, pp.13-54, 1998.

[39] J. Branke and S. Mostaghim, About selecting the personal best in multi-objective particle swarm optimization, *Proc. of the 9th International Conference on Parallel Problem Solving from Nature*, Reykjavik, Iceland, pp.523-532, 2006.

[40] S. Mostaghim and J. Teich, Strategies for finding local guides in multi-objective particle swarm optimization (MOPSO), *Proc. of IEEE Swarm Intelligence Symposium 2003*, Indianapolis, IN, USA, pp.26-33, 2003.

[41] J. E. Fieldsend and S. Singh, A multi-objective algorithm based upon particle swarm optimisation, and efficient data structure and turbulence, *Proc. of the 2002 U.K. Workshop on Computational Intelligence*, pp.34-44, 2002.

[42] E. Zitzler, K. Deb and L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, *Evolutionary Computation*, vol.8, no.2, pp.173-195, 2000.

[43] D. A. V. Veldhuizen and G. B. Lamont, Multiobjective evolutionary algorithm research: A history and analysis, *Technical Report*, Department of Electric and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH, pp.98-03, 1998.

[44] E. Zitzler and L. Thiele, Multiobjective evolutionary algorithms: A comparative case study and strengthen Pareto approach, *IEEE Transactions on Evolutionary Computation*, vol.3, no.4, pp.257-271, 1999.

[45] J. R. Schott, *Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization*, Master Thesis, Massachusetts Institute of Technology, Cambridge, MA, 1995.