A Fuzzy Grouping Genetic Algorithm for Care Task Assignment

M. Mutingi, C. Mbohwa

***Abstract*—The assignment of care tasks to nurses is often done manually in most hospitals. A high quality care task schedule is crucial for efficient and effective execution of nursing care duties. High quality schedules seek to satisfy patient preferences over time window for the care, schedule fairness among nurses, and management goals regarding care activity completion times and labor costs. This paper suggests a grouping genetic approach to care task scheduling in a hospital setting. By taking advantage of the group structure of the problem, the algorithm uses fuzzy evaluation techniques, permuting tasks across candidate nurse schedules and within each nurse schedule. Results of the computational experiments show that the proposed approach is effective.**

***Index Terms*—Care tasks, task assignment, fuzzy grouping genetic algorithm, fuzzy theory**

I. Introduction

CARE task assignment in hospitals involves allocation of nursing care activities to nursing staff on a daily basis, subject to hard and soft constraints regarding relationships between tasks and capacity limitation of nurses [1] [2]. The responsibility of nurses is to efficiently and effectively provide high quality care to patients. However, due to worldwide shortage of nurses and the ever-increasing pressure for high quality nursing care, care task assignment is a crucial but hard problem. In most hospitals, task assignment is done manually using spreadsheets based on pre-determined patient information regarding execution time intervals in which care activities should be done to patients [3]. Normally, tasks are assigned to nurses according to basic rules of the thumb [4]. However, this procedure may yield poor and unfair task schedules, leading to poor quality of service. Developing efficient and effective task scheduling methods is imperative.

High quality schedules should satisfy, as much as possible, the preferences and expectations of the patients, the nursing staff and the management. Care task schedules should ensure that the actual care execution times are as close as possible to the desired time windows pre-specified by patients. This leads to high patient satisfaction. In addition, care tasks should be assigned fairly among the available nurses; the goal is to balance, as much as possible, the individual workloads assigned to nurses. This ultimately leads to high worker moral, service efficiency, and job satisfaction. However, in satisfying the preferences of patients and nurses, the decision maker should consider management goals and expectations [5]. However, management goals are often qualitative and imprecise, adding to the complexity of the problem.

In the presence of imprecise and conflicting preferences and management goals, the use of conventional optimization methods such as linear programming, and basic dispatching heuristics such as earliest due date, slack, and first in first out, is limited [4] [5]. For instance, conventional dispatching rules disallow the use of multiple criteria in the scheduling process. Moreover, the rules have a rigid structure that excludes the use of other useful information that may be available. Thus, the care task assignment problem is characterized by complicating features:

(1) The presence of fuzzy staff preferences and wishes, such as fairness and equity on assigned workloads;
(2) The presence of fuzzy patient expectations and preferences on time windows and care due dates;
(3) The presence of imprecise management goals which are difficult to quantify; and,
(4) The need to find a judicious trade-off between conflicting goals of the problem.

Designing interactive metaheuristics to handle fuzzy goals and preferences is imperative. This will provide high quality task schedules that eventually lead to improved care worker satisfaction (job satisfaction), service efficiency, service quality, and business competitiveness. Incorporating fuzzy evaluation techniques into metaheuristic approaches is a viable and promising option. The purpose of this research is to develop a fuzzy heuristic approach to care task assignment in a hospital setting. Therefore, the specific objectives are:

(1) To describe the care task assignment problem;
(2) To propose a fuzzy grouping genetic algorithm; and,
(3) To provide illustrative examples, demonstrating the effectiveness of the algorithm.

The next section presents a brief description of the care task assignment. Section III presents a description of the proposed fuzzy grouping genetic algorithm. Computational experiments, results and discussions are presented in Section IV. Finally, conclusions and contributions are presented in Section V.

Manuscript received June 30, 2014; revised August 15, 2014.

M. Mutingi is a doctoral student with the Faculty of Engineering and the Built Environment, University of Johannesburg, Bunting Road Campus, P. O. Box 524, Auckland Park 2006, Johannesburg, South Africa (phone: +27789578693; e-mail: mmutingi@gmail.com).

C. Mbohwa is a professor with the Department of Quality and Operations Management, Faculty of Engineering and the Built Environment, University of Johannesburg, Johannesburg, Bunting Road Campus, P. O. Box 524, Auckland Park 2006, Johannesburg, South Africa (e-mail: cmbohwa@uj.ac.za).

Date	03Jan 2014								
Time	0800	0930	1000	1030	1100	1130	1200	1230	1300
Patient 1	Shampoo toilet Assist bath	Check gauze		BP check		BP check	Preprandial medicine	Serve, clear tray Clean teeth	
Patient 2	BP check	Anti-biotic		BP check Instil drops		Check compliance	Serve, clear tray Clean teeth		
…	…	…	…	…	…	…	…	…	…

Fig. 1. A typical example of a daily worksheet

II. The Care Task Assignment Problem

A. Problem Description

The health care task assignment problem (CTAP) is concerned with the allocation of a set of health care tasks to nursing staff so that patients can receive the required healthcare service [5] [6] [7]. The essence of the problem is that all the tasks must be assigned, subject to a set of constraints concerning care giver capacity, the nature of tasks and their precedence relationships. In most hospitals, care tasks are assigned based on pre-determined patient information on task duration and the time window during which specific tasks should be performed. The task assignment process is often carried out manually using spreadsheets. In practice, standard work procedures are recorded in manuals that contain key information on care tasks and the associated resources needed. Care tasks, for instance, assistance with meals, instillation of drops, and preparation of infusions, can be categorized into preparatory tasks, executions tasks, and clean-up tasks [4] [5]. Therefore, each care activity consists of these three types of tasks. Fig. 1 shows an example of a daily worksheet.

CTAP is analogous to the job dispatching or job shop scheduling problem. Similar to job dispatching, CTAP can be addressed using dispatching rule-based methods. In that context, appropriate priority rules, such as first-in-first out (FIFO) and earliest due date (EDD), can be applied. Deriving from this analogy, we identified a number of constraints, classified into hard constraints and soft constraints.

B. Problem Formulation

Table 1 presents the constraints associated with the CTAP problem [4] [5]. Hard constraints must always be satisfied, while soft constraints may be violated, but at a penalty cost. Hard constraints are concerned with task release time, task precedence, and staff capacity. For clarity of deliberation on the CTAP problem formulation, we define the following notations.

Notation:

u	Index for U activities, $u = 1,2,\ldots,U$
v	Index for V_u tasks in activity u, $v = 1,2,\ldots,V$
d_u	The desirable due date of activity u
r_u	The release time of activity u
t_{uv}	Starting time for care task v of activity u
$tc_{v1,v2}$	The changeover time from task $v1$ to $v2$
p_{uv}	The expected processing time of task v of activity u

1) Release Time Constraints:

The release time of a task pertains to the earliest time when that particular task is ready for execution.

$$t_{uv} \geq r_u \qquad \forall u, u = 1,2,...,U \quad \forall v, v = 1,2,...,V_u \qquad (1)$$

2) Precedence Constraints

Precedence constraints relate to the structural sequence of tasks that should be observed when executing specific tasks. It follows that succeeding care tasks cannot be handled until their predecessors are finished.

$$t_{u,v-1} + p_{u,v-1} \leq t_{uv} \quad \forall u, u = 1,2,...,U \quad \forall v, v = 1,2,...,V_u \qquad (2)$$

3) Capacity Constraints

Capacity constraints limit the number of tasks that a nurse can perform at any given time. In this case, we assume that a nurse can perform only one task at a time. Therefore,

$$t_{u1,v1} + p_{u1,v1} + tc_{v1,v2} \leq t_{u2,v2} \qquad (3)$$

where, $v1$ and $v2$ represent any two tasks from any activities to be performed by a specific nurse.

In addition to the three hard constraints above, soft constraints should be satisfied as much as possible. We identify three types of soft constraints concerned with due date, task changeover, and execution time (time window) of the tasks.

TABLE I
TYPICAL CARE TASK CONSTRAINTS

Constraint	Brief Description
Hard Constraints:	
1. Release Time Constraints	The first nursing task of a certain nursing activity cannot be handled until the release time of the activity.
2. Precedence Constraints	A nursing task cannot be handled until the previous task of the same nursing activity is finished.
3. Capacity Constraints	Nurses have the limited processing capacity to handle their work. In general, they can handle only one nursing task at a time.
Soft Constraints:	
4. Due Date Constraints	The last nursing tasks of a certain nursing activity should be finished before the due date of the nursing activity.
5. Transition Time Constraints	Some nursing tasks cannot be handled immediately after their previous task of the same nursing activity is finished.
6. Time Window Constraints	Some execution tasks of nursing activities should be handled within an expected execution time interval.

4) Due Date Constraints

The due date constraints ensure that the end time of each activity is as close as possible to the desirable due date of that activity. This restriction can be represented by the following expression;

$$t_{uv} + p_{uv} \le d_u \qquad \forall u, u = 1,2,...,U \quad \forall v, v = 1,2,...,V_u \tag{4}$$

5) Transitions Time Constraints

These constraints ensure that the changeover time between successive tasks is as much close to the desired time as possible. This implies that succeeding tasks should not be performed immediately after their predecessors, till a desired lapse of time (or transition time) $trans_{v1v2}$ is reached.

$$t_{uv2} \approx t_{uv1} + p_{uv1} + trans_{v1,v2} \quad \forall u, \forall v1, \forall v2 \tag{5}$$

where, $v1$ and $v2$ are tasks of the same activity u, and task $v2$ can only start after task $v1$ is completed, with a transition time $trans_{v1,v2}$.

6) Time Window Constraints

Time window constraints limit, as much as possible, the execution time of some care activities to be within the desired time window $[T_u^1, T_u^2]$, where T_u^1 and T_u^2 are the lower and upper bounds on the expected execution time of an activity u, respectively. For instance, lunch meals may be restricted to time window [1200, 1300] hrs. Therefore,

$$t_{uv} \ge T_u^1 \text{ and } t_{uv} + p_{uv} \le T_u^2, \qquad \forall u, \forall v \tag{6}$$

where, T_u^1 and T_u^2 are the lower and upper bounds on the execution time of activity u, respectively.

C. Problem Objectives

The objectives of the CTAP are (i) to maximize fairness in workload assignment, (ii) to minimize violation of soft constraints. Thus, the aim is to maximize the quality of the care schedule by finding a trade-off between these objectives. Clearly, the CTAP is a complex problem that is difficult to solve using conventional solution approaches. To this end, we present an enhanced fuzzy grouping genetic algorithm for interactive decision making for the problem.

III. A Fuzzy Group Genetic Algorithm Approach

Fuzzy grouping genetic algorithm (FGGA) is a development from grouping genetic algorithm [8]. It uses fuzzy theory to evaluate the performance of alternative solutions. FGGA takes advantage of the group structure of the problem. The algorithm and its elements, including chromosome representation, initialization, fuzzy fitness evaluation, and genetic operators, are presented in this section.

A. FGGA Coding Scheme

To enhance the performance of FGGA, a unique group coding scheme is developed to exploit the group structure of the problem. Let C = {1, 2, 3,…,V] be a chromosome representing a set of V tasks to be performed by I nurses. Then, the evaluation of C involves partitioning tasks along C into g groups such that all the hard constraints are satisfied and the violation of soft constraints is minimized. For instance, given 7 tasks (V = 7), and 3 nursing staff (I = 3), then the group structure of the problem is coded as shown in Fig. 2. The structure consists of two codes: code 1 represents the assignment of care workers w_1, w_2, and w_3, to groups of tasks {1,2}, {3,4,5}, and {6,7}, respectively. Genetic operators work on code 1, while code 2 records the position of the delimiter or frontier "|" which separates task groups.

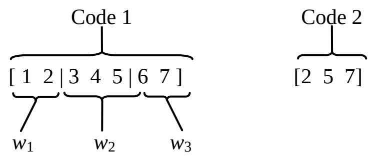

Fig. 2. FGGA chromosome coding scheme

B. Initialization

An initial population of candidate solutions is created by randomly assigning tasks to nurses. First, care tasks are arranged in ascending order of their expected start times. For each nurse, probabilistically allocate unassigned tasks, beginning with the earliest. By this procedure, the algorithm increases the likelihood of generating a good initial population of feasible solution candidates.

C. Fitness Evaluation

The fitness evaluation procedure determines the goodness of each candidate solution based on a combination of fuzzy evaluation functions. The functions should measure the relevant quality of the candidate solution, and capture the imprecise conflicting goals and constraints. The evaluation function F_t, at iteration t, should be a normalized function obtained from its n constituent normalized functions denoted by μ_h (h = 1,…,n). Therefore, we use a fuzzy multi-factor evaluation method,

$$F_t(s) = \sum_h w_h \mu_h(s) \tag{7}$$

where, s is a candidate solution at iteration t; and w_h denotes the weight of the function μ_h. The use of the max-min operator is avoided so as to prevent possible loss of vital information.

The goodness or quality of a solution is a function of how much it satisfies the preferences of nurses and patients, as well as management goals and choices. Due to ambiguity and imprecision of these preferences and goals, fitness is modelled as a normalized interval-valued fuzzy membership functions, as shown in Fig. 3.

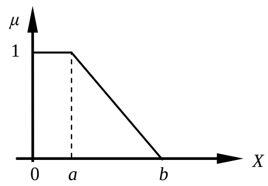

Fig. 3 Interval-valued linear membership function

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The satisfaction level is represented by a decreasing linear function where $[0,a]$ is the most desirable range, and b is the maximum acceptable. Therefore, the corresponding function is,

$$\mu_A(x) = \begin{cases} 1 & \text{If } x \le a \\ (b-x)/(b-a) & \text{If } a \le x \le b \\ 0 & \text{If otherwise} \end{cases} \quad (8)$$

1) Membership Function 1: Due Date

The completion time of every activity should, as much as possible, be within acceptable limits so as to maximize the quality of the care schedule. Let t_u and d_u be the completion time and due date of each activity u, respectively. Then the total variation of completion times from their respective due dates is given by the expression,

$$z_1 = \sum_u |t_u - d_u| \quad (9)$$

Assuming interval-valued membership function, we obtain,

$$\mu_1 = \mu_A(z_1) \quad (10)$$

2) Membership Function 2: Transition Time

For any activity u, assume that task $v2$ is supposed to start soon after task $v1$, with a transition time of $trans_{v1,v2}$. Then, the objective is to minimize the total variation z_2 from meeting this transition requirement, represented by,

$$z_2 = \sum_u \sum_{v1,v2 \in J_u} |t_{u,v1} + p_{u,v1} + trans_{v1,v2} - t_{u,v2}| \quad (11)$$

Therefore, assuming the interval-valued membership function, we obtain,

$$\mu_2 = \mu_A(z_2) \quad (12)$$

3) Membership Function 3: Time Windows – Earliness

The goodness of a candidate care schedule can be measured in terms of total earliness. Let the total earliness be denoted by z_3. Then,

$$z_3 = \sum_u \sum_{v \in J_u} \max(0, T_u^1 - t_{uv}) \quad (13)$$

Here, J_u is a set of tasks from activity u; T_u^1 is the lower bound of the time window of activity u. Assuming that e follows a trapezoidal linear membership, the fuzzy membership function for earliness can be expressed as follows;

$$\mu_1 = \mu_A(z_3) \quad (14)$$

4) Membership Function 4: Time Window - Lateness

Apart from earliness, the goodness of a candidate care schedule can also be measured in terms of total lateness or tardiness. Therefore, the objective is to minimize total tardiness z_4 given by,

$$z_4 = \sum_u \sum_{v \in J_u} \max(0, t_{uv} + p_{uv} - T_u^2) \quad (15)$$

Here, T_u^2 is the upper bound on time window of activity u, and J_u is a set of tasks in activity u. Assume that z_4 follows a trapezoidal linear membership. Then the fuzzy membership function for tardiness can be expressed as follows;

$$\mu_4 = \mu_4(z_4) \quad (16)$$

5) Membership Function 5: Workload Fairness

Let h_i denote the workload of nurse i, and a be the average workload. The objective is to minimize workload variation z_5 is given by,

$$z_5 = \sum_i |h_i - a| \quad (17)$$

Here, the workload h_i is given by the expression,

$$h_i = \sum_u \sum_v p_v x_{iv} \qquad \forall i, i = 1, 2, ..., N \quad (18)$$

where, x_{iv} is a binary variable representing whether or not task v is assigned to nurse i. We use a fuzzy membership function μ_5 of the form,

$$\mu_5 = \mu(z_5) \quad (19)$$

D. Selection and Crossover

The selection operator selects the best performing chromosomes into a mating pool, called *tempp*. Among various selection mechanisms, remainder stochastic sampling without replacement is the most effective (Goldberg, 1989; Holland. 1975), and therefore was adopted in this study. Each chromosome c is selected and stored in *tempp* according to its expected count e_s,

$$e_s = \frac{F(s)}{1/p \sum_{s=1}^{p} F(s)} \quad (20)$$

where, p is the population size; F_s (s = 1,…,p) is the fitness function of the s^{th} chromosome.

According to this strategy, each chromosome receives copies equal to the integer part of e_s, plus additional copies obtained by using the fractional part of e_s as a success probability of getting an additional copy of chromosome s. The best performing candidates are selected with higher probability into *tempp*.

Crossover is a stochastic mechanism by which selected chromosomes mate to produce new offspring, called *selection pool*. The mechanism enables FGGA to explore

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

unvisited regions in the solution space. Groups of genes in the selected chromosomes are exchanged at a probability *pcross*. First, a crossover point between 1 and g is randomly generated, where g is the number of groups. Second, the groups on the right of the crossover point are swapped. Third, the offspring are repaired as necessary. The process is repeated till the desired pool size, *poolsize*, is achieved. Fig. 4 shows a crossover of parent chromosomes P_2 and P_2. Offspring O_1 and O_2 are repaired to O_1' and O_2'.

Parents:	Offspring:	Repaired:
P_1: [2 5 | 3 4 1 | 6]	O_1: [2 5 | 3 1 | 6]	O_1': [2 5 | 3 1 | 4 6]
P_2: [5 6 | 3 1 | 4 2]	O_2: [5 6 | 3 4 1 | 4 2]	O_2': [5 6 | 3 4 1 | 2]

Fig. 4. An example of crossover and repair mechanisms

After crossover, some of the genes may appear in more than one group, while others may be missing. Such offspring are repaired by eliminating duplicated genes on either side of the crossover point, and then inserting missing genes into those groups with the least workload. Group coding takes advantage of the group structure to generate new offspring.

E. Mutation

Mutation is applied to every new chromosome in two forms: swap mutation and shift mutation. Swap mutation exchanges genes between two groups in an individual chromosome, while shift mutation moves a randomly chosen frontier between two adjacent groups by one step to the right or to the left. Thus, the mutation operator provides FGGA with local search capability, a phenomenon called intensification. Fig. 5 provides an illustration of swap and shift mutation mechanisms, respectively.

	Swap Mutation	Shift Mutation
Before mutation:	[1 2 | 3 4 5 | 6]	[1 2 | 3 4 5 | 6]
After mutation:	[1 2 | 6 4 5 | 3]	[1 2 | 3 4 | 5 6]

Fig. 5. An illustration of swap and shift mutation

F. Inversion and Diversification

The population may prematurely converge to a particular solution, thus, population diversity should be controlled. Inversion is a stochastic operator that restructures the genes of a chromosome in the reverse order, e.g., chromosome [21|5|34] may be transformed to [34|5|21]. To check diversity, we define an entropic measure h_k for task v;

$$h_v = \sum_{j=1}^{n} \frac{(x_{jv}/p)\cdot \ln(x_{jv}/p)}{\ln V} \tag{21}$$

where, x_{jv} represents the number of chromosomes in which task k appears in position j of chromosomes in the current population; V is the number of tasks. Therefore, diversity h is given by,

$$h = \sum_{k=1}^{n} h_k / n \tag{22}$$

In this application, the inversion operator is applied whenever diversity falls below a threshold value, h_d, while preserving best performing candidates.

G. The Overall FGGA Algorithm

The overall algorithm incorporates the above operators, beginning with the selection of suitable input parameters. The selected input parameters were: crossover probability (0.35), mutation probability (0.01), and inversion probability (0.04). An initial population, $P(0)$, is generated randomly by random assignments of clients to care givers. The algorithm then proceeds into an iterative loop involving selection, crossover, mutation, inversion, and until termination condition is reached (maximum number of iterations T). Fig. 6 presents the overall structure of the proposed FGGA.

```
The Overall FGGA Procedure
BEGIN
    Input: FGGA parameters; t = 0;
    Initialize population, P(0);
    REPEAT
        Selection:
            Evaluate P(t);
            Create temporal population, tempp(t);
        Group crossover:
            Select 2 chromosomes from tempp(t);
                Apply crossover operator
                Repair if necessary;
            Mutation:
                Mutate P(t);
                Add offspring to newpop(t);
            Replacement:
                Compare successively, spool(t) and oldpop(t) strings;
                Take the ones that fare better;
                Select the rest of the strings with probability 0.55;
            Diversification:
                Calculate population diversity H;
                WHILE (h < h_d)
                    diversify P(t);
                    calculate h;
                END WHILE
                Evaluate P(t);
        New population:
                oldpop(t) = newpop(t);
                Advance population, t = t + 1
        UNTIL (t ≥ T)
    END
```

Fig. 6. The overall FGGA pseudo-code

We present illustrative examples, computational results, and relevant discussions in the next section.

IV. Computational Experiments And Results

A. Computational Experiments

To test the proposed method, we randomly generated the data for the care task assignment problem. We assumed a hospital department with 30 patients 5 nurses working in a day shift from 8:00 am to 5:00 pm. We further assumed a normal distribution for the processing times, p_{uv}, according to the expression, $p_{uv} \in [\bar{p}_{uv} - \varepsilon \cdot \bar{p}_{uv}, \bar{p}_{uv} + \varepsilon \cdot \bar{p}_{uv}]$, where $\varepsilon \in [0,1]$. As such, we randomly generated 10 problems with different processing times by setting $\varepsilon = 0.5$. The release times r_u for each activity u, time windows for specific tasks, and the due dates for different activities were also created randomly between 8:00 am an 5:00 pm.

The algorithm was coded in Java™ 7, Standard Edition,

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Windows 7 operating system on a PC running on an Intel Pentium 3.0 GHz, and 4GB RAM. The performance of FGGA was compare against particle swarm optimization (PSO) and genetic algorithm (GA) which were developed in this study [8] [9].

B. Results and Discussions

Fig. 7 shows an illustration of the transcription of the fitness values for a maximum of 500 iterations. The comparative performance of FGGA against PSO and GA show that FGGA outperforms the two competitive algorithms in terms of efficiency and the final fitness value.

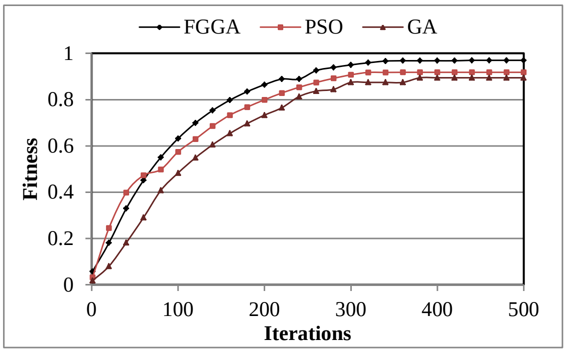

Fig. 7. Comparative performance of FGGA, PSO and GA

TABLE II
COMPARATIVE ANALYSIS

Algorithm	Average CPU time (sec)	Search Success Rate	Solution, F_t
FGGA	4.6	100.0%	1.00
PSO	5.5	93.3%	1.00
GA	6.2	90.0%	1.00

The algorithms were tested for search success rate based on a hypothetical problem consisting of 4 nurses and 24 tasks, with a known optimal solution. Each algorithm was run 20 times, recording the average computation (CPU) time, and the search success rate. Table II shows the results of the experiment. All the algorithms obtained the desired optimal solution. In terms of search success rate, FGGA rated 100%, followed by PSO rated 93.33%, while GA rated 90%. Regarding computational efficiency, FGGA performed the best with CPU time 4.6 seconds. On the other hand, the average CPU times for PSO and GA were 5.5 sec and 6.2 sec, respectively. Therefore FGGA has a higher potential for efficient and effective performance, even on larger scale problems.

V. CONCLUSION

Care task assignment is a common problem in hospitals, concerned with finding the best way to allocate care tasks to a limited pool of nurses, so that all tasks are performed as timely as possible, nurse workloads are assigned fairly, transition times between tasks are satisficed as much as possible, and expert choices are taken into account. The higher the satisfaction level of these requirements, the higher the quality of the task schedule. This is a hard problem that demands interactive fuzzy heuristic methods. This paper presented a fuzzy group genetic algorithm to solve the problem. By exploiting permutations of groups of tasks across candidate task schedules and within each candidate schedule, using enhanced heuristic operators, the algorithm can address the problem efficiently. The approach provides useful contributions to researchers and practitioners in healthcare.

A. Contributions to Knowledge

The proposed algorithm contributes to knowledge in flexible, adaptable and interactive heuristic optimization methods. By realizing the need to holistically satisfy the patient, the healthcare worker, and the management, this research provides a judicious trade-off approach by which the three players in a healthcare system can be satisfied, with potential long-term benefits. Moreover, the problem can be modelled with more realism, considering fuzzy expert choices of the decision maker. The method presents in-built heuristic techniques that exploit the group structure of the problem to handle large-scale problems efficiently. Thus, the proposed algorithm is an invaluable addition to the body of knowledge in healthcare operations management.

B. Contributions to Practice

The practicing decision maker can benefit from the suggested approach to the CTAP in a number of ways. The algorithm provides an opportunity to use weights to interactively incorporate the decision maker's preferences and choices. In practice, decision makers appreciate the use of an interactive decision support that provide a list of good alternative solutions from which the most appropriate decision can be chosen, taking into account other practical considerations. In this view, expert knowledge can be incorporated into the decision process, unlike when using prescriptive optimization methods. Overall, the proposed algorithm a viable tool in care task assignment.

REFERENCES

[1] I. Vermeulen, S. Bohte, K. Somefun, and H. L. Poutre, "Improving patient activity schedules by multi-agent Pareto appointment exchanging", Proc. of the 8th IEEE Conference on E-Commerce Technology (CEC' 06), 2006.

[2] T.O. Paulussen, N.R. Jennings, K.S. Decker, A. Heinzl "Distributed patient scheduling in hospitals", *Proc. of the 18th International Joint Conference on Artificial Intelligence*, pp. 1224-1229, 2003.

[3] M. Cheng, H.I. Ozaku, N. Kuwahara, K. Kogure, J. Ota, "Simulated annealing algorithm for scheduling problem in daily nursing cares," *Proc. of the IEEE International Conference on Systems, Man and Cybernetics*, SMC, pp. 1681-1687, October 2008, 2008,.

[4] M. Cheng, H.I. Ozaku, N. Kuwahara, K. Kogure, J. Ota, 'Nursing care scheduling problem: Analysis of staffing levels', *Proc. of the 2007 IEEE International Conference on Robotics and Biomimetics*, Sanya, China, December 15-18, 2007.

[5] M. Mutingi, C. Mbohwa, "Care task assignment: A simulated metamorphosis approach (Working paper)," unpublished, 2014.

[6] J.F. Bard, H.W. Purnomo, "Real-time scheduling for nurses in response to demand fluctuations and personnel shortages" *Proceedings of the 5th International Conference on Scheduling: Theory & Applications (MISTA'05)*, pp. 397-406, 2005.

[7] L.H. Aiken, S.P. Clarke, D.M. Sloane, "Hospital staffing, organization, and quality of care: cross-national findings", *Int Journal for Quality in Health Care*, vol. 14, no. 1, pp. 5-13, 2002.

[8] M. Mutingi, C. Mbohwa, "Enhanced group genetic algorithm for the heterogeneous fixed fleet vehicle routing problem", *IEEE IEEM Conference on Industrial Engineering and Engineering Management*, Hong Kong, pp. 207-2011, Dec 2012.

[9] M. Mutingi, C. Mbohwa, "Home Healthcare Staff Scheduling: A Clustering Particle Swarm Optimization Approach", *Proc. of the 2014 International Conference on Industrial Engineering and Operations Management*, Indonesia, pp. 303-312, Jan. 7 – 9, 2014

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)