

Home Healthcare Worker Scheduling: A Group Genetic Algorithm Approach

M. Mutingi and C. Mbohwa

Abstract—Home healthcare worker scheduling is a hard combinatorial problem concerned with the allocation of care tasks to healthcare givers at a minimal cost while considering healthcare service quality by striving to meet the time window restrictions specified by the patients. This paper proposes a group genetic algorithm (GGA) for addressing the scheduling problem. The approach utilizes the strengths of unique group genetic operators to effectively and efficiently address the group structure of the problem, providing good solutions within reasonable computation times. Computational results obtained show that the GGA approach is effective.

Index Terms—Home healthcare, group genetic algorithm, Multi-objective optimization, Staff scheduling

I. INTRODUCTION

HOME health care services provide health care to people in their community homes, in accordance with their specific health needs [1]. The overall goal is to provide satisfactory care and assistance to patients who need special care at their homes, that is, patients with physical or mental challenges, patients with terminal or acute illness, patients in need of post-operation or post-hospitalization treatment, as well as the elderly people with various healthcare needs [2] [3]. Healthcare professionals are essential for providing services such as therapy services, medical and social services, and house cleaning. The need for home care is accelerated by a number of factors of common occurrence in various communities all over the globe. These factors include ageing population, ever-increasing chronic pathologies, increasing innovative technologies, over and above pressures from governmental authorities for improved health care services. Communities continue to call for continuous improvement in healthcare service quality, which has become the central objective in most healthcare organizations [3]. Consequently, home health care services continue to increase in size leading to difficulties in staff scheduling as well as task assignment to specific healthcare professionals or care givers that have to visit the patients at their homes at specific time windows preferred by the patients.

Manuscript received 23 March, 2013.

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

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

Without appropriate decision support tools, scheduling and task assignment of care givers is a complex and time-consuming responsibility. Health care workers travel for some distance to deliver care to their assigned clients at their homes within a specified time window, and then return to their original workplace after finishing all their assigned visits. However, all the tasks assigned to the care giver have to be completed within the capacity limit of the care giver; for instance, most healthcare services limit the capacity of a single giver to about 8 hours per day [3]. The time window for each duty or activity is dependent on the specific needs of the client according to the pre-assessment of the client's health condition. This implies that the time windows of critical tasks tend to be tighter than those of low-level criticality. In addition to these constraints, home care scheduling can be time-consuming, especially when taking into account the shift preferences of care givers [3] [4] [5]. Efficient and robust tools are essential for effective home care scheduling.

In regards to the above highlighted issues, the purpose of this research is to present an efficient group genetic algorithm (GGA) for scheduling the dispatch of health care givers to clients, while considering time and capacity constraints. To this end, the specific objectives of this research are;

- (1) To describe the home care worker scheduling problem as an extension of the vehicle routing problem with time windows;
- (2) To develop a GGA approach for addressing the home care worker scheduling problem;
- (3) To carry out illustrative computational tests, so as to show the utility of the proposed GGA approach.

The GGA approach is designed to assign duties to care givers on a daily basis so as to minimize the total costs incurred in terms of distance travelled and the cost associated with the violation of time window specifications. The ultimate advantages of efficient home care scheduling are vast; these include the following:

- travelling costs of home care workers are reduced significantly;
- worker utilization is improved, which leads to reduced workforce needs;
- improved care service, satisfying customer requirements with specified time windows; and,
- reduced schedule construction time, freeing up manager's time for strategic roles.

Homecare worker scheduling is a hard combinatorial problem that demands the application of efficient and

effective solution methods such as expert systems, metaheuristics, other intelligent approaches, or a combination of these [3] [4] [5]. Therefore, GGA is a potential decision support tool in home healthcare operations, especially for scheduling homecare worker operations.

The remainder of this paper is organized thus: The next section briefly describes the homecare worker scheduling problem. This is followed by a description of the proposed GGA methodology, showing the unique genetic operators implemented in this paper. An illustrative computational example is then provided. Results and discussions are then presented, based on the case example. Finally, conclusions and further research prospects are presented.

II. PROBLEM DESCRIPTION

The home healthcare worker scheduling problem with time windows (see Fig. 1) can be described as follows [3] [4] [5]: Consider a community healthcare centre with m care givers to visit n clients, where each care giver k ($k = 1, 2, \dots, m$) is supposed to complete serving client j ($j = 1, 2, \dots, n$) within a given time window defined by earliest start time and latest start time, e_j and l_j , respectively. The context of the problem is analogous to the vehicle routing problem with time windows [6] [7]. The aim of the management is to minimize the total cost incurred by each care giver in travelling from the point of origin to the clients and back to the point of origin. Furthermore, if the care giver arrives at the client earlier than e_j or later than l_j , then a penalty cost is incurred in each case. Let a_j denote the time when a care giver reaches client j , and k_e and k_l denote the unit penalty costs incurred when the care giver arrives too early or too late, respectively. This implies that the functions, $\max[0, e_j - a_j]$ and $\max[0, a_j - l_j]$ have to be minimized. The aim is to maximize client satisfaction by constructing schedules with minimum violation of the preferred time window constraint as specified by the client. On the other hand, the decision maker seeks to improve the schedule quality by developing the most equitable schedule allocations. Therefore, the individual workloads or worker capacity should, as much as possible, be kept within the limits of the workers preferences.

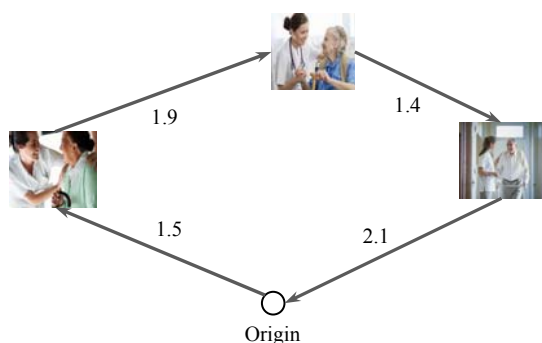


Fig. 1 An illustrative of a home care worker schedule

III. A GROUP GENETIC ALGORITHM APPROACH

The GGA approach, originally proposed by Falkenauer [8] is a development of the basic genetic algorithm (GA) [9] [10], for addressing grouping problems. We present the proposed GGA procedure and its elements, including chromosome coding, initialization, and genetic operators. As shown in Fig. 2, the general structure of GGA is similar to the basic GA. However, the internal mechanisms of the coding scheme, and the implementation of the genetic operators is unique to GGA.

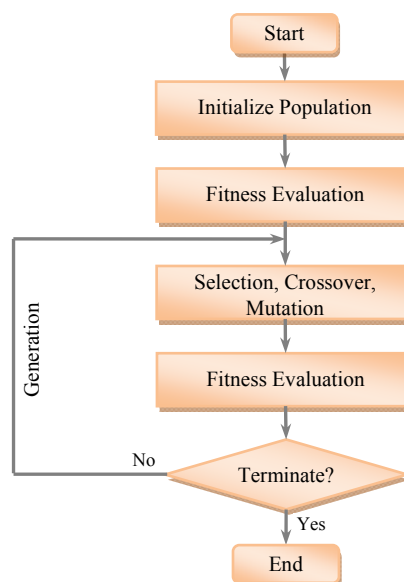


Fig. 2 Basic group genetic algorithm structure

A. GGA Coding Scheme

The performance of GGA is strongly influenced by the structure of the genetic coding applied [11] [12]. We develop a unique coding scheme which exploits the group structure of the scheduling problem. In this vein, let $C = [1, 2, 3, \dots, n]$ be a chromosome representing a set of n clients to be visited by m care givers. The evaluation of C involves partitioning clients along C into m groups such that the cumulative cost incurred is minimized, and the cumulative load for each group does not exceed the care giver capacity limit.

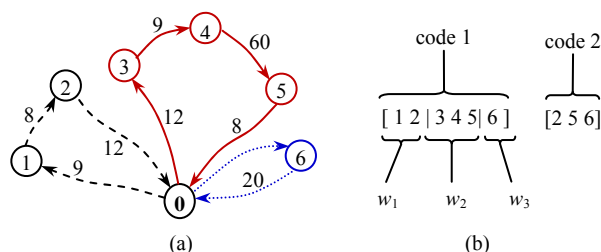


Fig. 2 Group genetic algorithm chromosome coding scheme

Fig. 2 gives an example of a healthcare manpower system consisting of 3 workers and 6 clients; part (a) provides the network structure of the problem while part (b) illustrates the group structure of the chromosome consisting of two codes. Code 1 represents the assignment of care workers w_1 ,

w_2 , and w_3 , to groups of clients $\{1,2\}$, $\{3,4,5\}$, and $\{6,7\}$, respectively. Code 1 is the actual group structure upon which the genetic operators act. Code 2 denotes the last position of each client group, that is, it records the position of the delimiter or frontier “|” which separates client groups. The total cost in terms of distance travelled is $(12 + 8 + 9) + (12 + 9 + 6 + 8) + (20 + 20) = 104$.

B. Initialization

An initial population of the desired size, p , is randomly created by random assignments of clients to care givers. First, arrange the care duties in ascending order of their start times. In case of a tie, rank the duties according to their activity duration. For each care giver, assign a duty at a probability b , starting from the earliest. From the unassigned set of duties, assign duties beginning from the earliest. This procedure increases the likelihood of the initialization process to generate initial feasible solutions.

C. Fitness Evaluation

The fitness function evaluation calculation is done after the schedules have been constructed. Let $a(i,j)$ represent a feasible trip, where the care giver departs from point of origin 0 and visits nodes $i+1, i+2, \dots, j-1$, and j , consecutively. Then, the cost of each trip, c_{ij} is estimated in terms of the distance from the care giver’s point of origin to the first client, the distances between successive clients, the distance from the last client back to the care giver’s point of origin, and the penalty cost incurred due to violation of time window preferences;

$$c_{ij} = v \left(d_{0,i+1} + \sum_{h=i+1}^{j-1} d_{h,h+1} + d_{j,0} \right) + k_e \sum_{i=1}^j \max(0, l_i - a_i) + k_l \sum_{i=1}^j \max(0, e_i - a_i) \quad (1)$$

Here, d_{ij} denotes the distances between successive clients i and j ; v is the variable cost per unit distance travelled; k_e and k_l denote, respectively, the unit penalty costs when a care giver reaches the client too early or too late; e_j and l_j are the earliest start time and latest start times preferred by client j ; and a_j denotes the actual time when a care giver reaches client j .

The objective function of each schedule or chromosome is equivalent to the sum of the costs of all the trips in the schedule. Our GGA maps the objective function to a fitness function f_k as follows,

$$f_k(t) = \max \left[0, g^m(t) - g_k(t) \right] \quad (2)$$

where, $g_k(t)$ is the objective function of chromosome k at time t ; and g^m is the maximum objective function in the current population.

D. Selection

The purpose of the selection operator is to select the best performing chromosomes into a mating pool, which is called *temppop*. A number of selection strategies have been suggested by Goldberg [9], including deterministic sampling, remainder stochastic sampling with replacement, remainder stochastic sampling without replacement, and

stochastic tournament. In this study, we adopted the remainder stochastic sampling without replacement method. According to this strategy, each chromosome k is selected and stored in the mating pool according to its expected count e_k , which is calculated according to the following expression;

$$e_k = \frac{f_k}{(1/p) \sum_{k=1}^p f_k} \quad (3)$$

where, f_k is the fitness function of the k^{th} chromosome.

Each chromosome receives copies equal to the integer part of e_k , that is, $[e_k]$, plus additional copies obtained with a success probability equal to the fractional part $\text{frac}(e_k)$ is treated as a success probability of obtaining additional copies of chromosome k into the mating pool. Thus, the best performing candidate solutions are selected into the mating pool with higher probability.

E. Crossover

Crossover is a mechanism through which selected chromosomes mate to produce new offspring, called selection pool. This enables GGA to explore unvisited regions in the solution space, which essentially provides the algorithm with explorative search abilities. Groups of genes in the selected chromosomes are exchanged at a probability p_{cross} . First, randomly generate a crossover point c between 1 and g , where g is the number of groups, that is, $c = \text{random}(1,g)$. Second, swap all the groups on the right of the crossover point. Third, repair the offspring, if necessary. This process is repeated till the desired pool size, *poolsize*, is achieved. Fig. 3 illustrates the crossover mechanism using parent chromosomes P_1 and P_2 . Upon crossover, offspring O_1 and O_2 are produced, which are necessarily repaired to produce O_1' and O_2' .

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

Fig. 3 An illustration of the crossover operator and repair mechanism

After crossover, some genes may appear in more than one group, while others may be missing. These offspring are repaired by (i) eliminating duplicated genes to the left and right of the crossover point, and (ii) inserting missing genes into the groups with the least loading. Here, group coding takes advantage of the group structure to generate new offspring. The mutation operation follows the crossover operator.

F. Mutation

Mutation is applied to every new chromosome using two mutation operators, namely, swap mutation and shift mutation. The swap mutation swaps genes between two groups in an individual chromosome, while the shift mutation works by shifting a randomly chosen frontier between two adjacent groups by one step, either to the right or to the left. In retrospect, the mutation operator essentially provides GGA with local search capability, a phenomenon

called intensification. However, the shift mutation is a more localized search operation than the swap mutation. Fig. 4 (a) and (b) provides an illustration of the swap and shift mutation mechanisms, respectively.

Before mutation : [5 2 | 4 3 1 | 6] [5 2 | 4 3 1 | 6]
After mutation : [5 2 | 6 3 1 | 4] [5 2 | 4 3 1 | 6]
(a) (b)

Fig. 4 An illustration of the swap and shift mutation operators

G. Inversion and Diversification

As iterations proceed, the population converges to a particular solution. However, population diversity has to be controlled in order to avoid premature convergence before an optimal solution is obtained, a process called genetic drift. Inversion is a genetic mechanism by which genes of a chromosome are rearranged in the reverse order, at a very low probability, for the purpose of improving the diversity of the population at each generation (iteration). The inversion operation can be illustrated based on the chromosome [1 2 | 4 | 3 5 6] as follows;

Before inversion : [1 2 | 4 | 3 5 6]
After inversion : [6 5 3 | 4 | 2 1]

To check diversity, first, we define an entropic measure H_i for each client i ;

$$h_i = \sum_{j=1}^n \frac{(x_{ij}/p) \cdot \ln(x_{ij}/p)}{\ln(n)} \quad (4)$$

Here, x_{ij} is the number of chromosomes in which client i is assigned position j in the current population; n is the number of clients. Therefore, diversity H can be defined according to the expression,

$$h = \sum_{i=1}^n h_i / n \quad (5)$$

In this connection, inversion is applied whenever diversity falls below a threshold value, h_d . The best performing candidates should always be preserved by comparing the diversified and undiversified populations and keeping a pre-specified number of the best three candidates in the population.

H. GGA Overall Algorithm

The overall GGA pseudo code incorporates the operators described in previous sections. The algorithm begins with the selection of suitable input genetic parameters. The selected input genetic parameters in this study were as follows: crossover probability (0.4), mutation probability (0.01), and inversion probability (0.05). An initial population, $P(0)$, is then generated randomly by random assignments of clients to care givers. The algorithm then proceeds into an iterative loop involving selection, group crossover, mutation, replacement strategy, inversion and diversification, population advancement, and termination condition test which allows successive iterations up until the number of iterations reaches a pre-specified maximum T .

Fig. 6 outlines the overall structure of the proposed GGA approach and all the constituent genetic operators as described in the previous sections.

```

BEGIN
  1. Input: GGA parameters;  $t = 0$ ;
  2. Initialize population,  $P(0)$ ;
  REPEAT
  4. Selection:
    Evaluate  $P(t)$ ;
    Create temporal population,  $tempopop(t)$ ;
  5. Group crossover:
    Select 2 chromosomes from  $tempopop(t)$ ;
    Apply crossover operator; Repair if necessary;
  6. Mutation:
    Mutate  $P(t)$ ;
    Add offspring to  $newpop(t)$ ;
  7. Replacement strategy:
    Compare successively,  $spool(t)$  and  $oldpop(t)$  strings;
    Take the ones that fare better;
    Select the rest of the strings with probability 0.55;
  8. Diversification:
    Calculate population diversity  $H$ ;
    IF ( $H < h_d$ ) THEN diversify till  $H \geq h_d$ ;
    Evaluate  $P(t)$ ;
  9. New population:
     $oldpop(t) = newpop(t)$ ;
    Advance population,  $t = t + 1$ 
  UNTIL ( $t \geq T$ )
END
    
```

Fig. 5 Pseudo-code for the overall GGA approach

In the next section, we present an illustrative example, the computational results, together with the relevant discussions.

IV. ILLUSTRATIVE EXAMPLE AND RESULTS

A. Illustrative Example

For the purpose of illustration, we adapt a problem instance presented in [6] [7]; Assume that there are eight healthcare tasks, with their respective execution time t_i , time windows $[e_j, l_j]$ as shown in Table I. The distance d_{ij} from origin 0 to each task point and between all other adjacent task points are provided in Tables II. We further assume that the unit travel cost is 1, and the penalty cost for violating any time window preference is $k_e = k_l = 50$.

TABLE I
TASK DURATIONS AND TIME WINDOWS

Task	1	2	3	4	5	6	7	8
t_i	1	2	1	3	2	2.5	3	0.8
$[e_j, l_j]$	[1,4]	[4,6]	[1,2]	[4,7]	[3,5.5]	[5,8]	[5,8]	[1.5,4]

TABLE II
DISTANCE BETWEEN TASK POINTS

d_{ij}	0	1	2	3	4	5	6	7	8
0	0	40	60	75	90	200	100	160	80
1	40	0	65	40	100	50	75	110	100
2	60	65	0	75	100	100	75	75	75
3	75	40	75	0	100	50	90	90	150
4	90	100	100	100	0	100	75	75	100
5	200	50	100	50	100	0	70	90	75
6	100	75	75	90	75	70	0	70	100
7	160	110	75	90	75	90	70	0	100
8	80	100	75	150	100	75	100	100	0

B. Results and Discussions

The computational results for the problem are presented in Table III. The solution presented corresponds to the best known result; with a total travel cost for the solution is 910. Our proposed GGA approach was able to obtain the best known solution, which demonstrates the effectiveness of the GGA metaheuristic method.

TABLE III
GGA SOLUTION

Homecare worker	Route/Path
1	0 → 8 → 5 → 7 → 0
2	0 → 3 → 1 → 2 → 0
3	0 → 6 → 4 → 0
Total Cost	910

Table IV shows the comparative analysis of the results obtained. Comparison was done against the results previously obtained using competitive algorithms, that is, basic GA, particle swarm optimization (PSO), parallel PSO, and hybrid PSO. The average computation time for GGA was 3.5, which is less than other algorithms. Moreover, the search success rate for GGA 98%, which is higher than other algorithms: 46% for PSO, 24% for basic GA, 72% for parallel PSO, and 97% for Hybrid PSO, as shown in Table IV. Therefore, the proposed method is effective and efficient in solving homecare worker scheduling problem.

TABLE IV
COMPARATIVE RESULTS

Approach	Cost	Search Success Rate (%)	Average Search Time(s)
GA	993.6	24	11.0
PSO	940.5	46	6.00
Parallel PSO	923.8	72	4.00
Hybrid PSO	914.0	97	4.00
GGA	910.0	98	3.50

V. CONCLUSION AND FURTHER RESEARCH

Home healthcare worker scheduling is a complex problem concerned with meeting the needs of patients at their homes at specific preferred time windows. We proposed a group genetic algorithm approach for solving typical problems, providing an illustrative example. The results demonstrated that, compared to other related approaches, GGA is an efficient and effective algorithm for addressing the home-based care problem. The approach is a potential tool for developing decision support systems for managers in healthcare systems concerned with home-based care community services. In our further research, we intend to apply the proposed algorithm to a typical home-based care centre in South Africa.

ACKNOWLEDGMENT

The authors would like to appreciate the reviewers for their constructive comments on the earlier version of this paper.

REFERENCES

- [1] R.B. Bachouch, A.G., Liesp. L. Insa, S. Hajri-Gabouj, An optimization model for task assignment in home healthcare. *IEEE Workshop on Health Care Management (WHCM)*, pp.1-6, 2010.
- [2] C. Akjiratikarl, Yenradee, P. and P.R. Drake, "PSO-based algorithm for home care worker scheduling in the UK. *Computers & Industrial Engineering*, vol. 53, pp.559-583, 2007.
- [3] P. Drake and B. M. Davies, "Home care outsourcing strategy," *Journal of Health Organization and Management*, vol. 20 no. 3, pp.175-193, 2006.
- [4] S. Bertels, and T. Fahle., "A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem", *Computers & Operations Research*, vol. 33, no. 10, pp.2866-2890, 2006.
- [5] S. Braysy, T. A. Bertels, Fahle, "Hybrid setup for a hybrid scenario: combining heuristics for the home health care problem", *Computers and Operations Research*, vol. 33, no. 10, pp.2866-2890, October 2006.
- [6] L. Xiaoxiang, J. Weigang, X. Jianwen, "Vehicle Routing Problem with Time Windows: A Hybrid Particle Swarm Optimization Approach", *IEEE 2009 Fifth International Conference on Natural Computation*, pp.502-506.
- [7] J. Li,Y-H. GUO Yao-huang. "Dispatch optimizes theory and methods for logistic distribution vehicle". Beijing: Chinese Commodity Publish House, 2001.
- [8] E. Falkenauer. "The grouping genetic algorithms - widening the scope of the GAs," *Belgian Journal of Operations Research, Statistics and Computer Science*, vol. 33, pp.79-102, 1992.
- [9] D. E. Goldberg. *Genetic Algorithms: In Search, Optimization & Machine Learning*, Addison-Wesley, Inc., MA, 1989.
- [10] J. H. Holland. *Adaptation in Natural and Artificial System*, University of Michigan Press, Ann Arbor, MI, 1975.
- [11] M. Mutingi and 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, 2012 (in press).
- [12] E.V.G. Filho and A.J. Tiberti. "A group genetic algorithm for the machine cell formation problem," *International Journal of Production Economics*, vol. 102, pp.1-21, 2006.