Grouping Genetic Algorithm for Industrial Engineering Applications

[1]Michael Mutingi, [1]Herbert Mapfaira and [2]Partson Dube
[1]Industrial Engineering, University of Botswana, Gaborone, Botswana
[2]School of Mechanical and Industrial Engineering, University of Johannesburg, South Africa
email: michael.mutingi@mopipi.ub.bw

Abstract
Industry is inundated with grouping problems concerned with formation of groups or clusters of system entities for the purpose of improving the overall system efficiency and effectiveness. Various extant grouping problems include cell formation problem, vehicle routing problem, bin packing problem, truck loading, home healthcare scheduling, and task assignment problem. Given the widespread grouping problems in industry, it is important to develop a tool for solving such problems from a common view point. This paper seeks to identify common grouping problems, identify their common grouping structures, present an outline of group genetic algorithm (GGA), and map the problems to the GGA approach. The practicality of the GGA tool in is highly promising in Industrial Engineering applications.

Keywords
Grouping problems, grouping genetic algorithms, genetic algorithms, metaheuristics, Industrial engineering

1. Introduction

Manufacturing and service industries are inundated with problems that require formation of groups or clusters of system entities with the aim of achieving a certain goal, typically to optimize the system efficiency and effectiveness [1]. For instance, in logistics and transport management, it is often desired to minimize transportation costs, number of vehicles used, and customer waiting times [2]. This can be achieved by optimizing the assignment of groups of customers to be visited by each vehicle or driver. As such, it is important how the grouping of customers is done, considering the size, type and capacity of the available vehicles. In the same vein, it may be desirable in container loading to stack the freight in an optimal way so as to minimize loading costs. Similarly, when assigning tasks to workers, it is crucial how groups of tasks may be formed and assigned to workers in an optimal manner. Furthermore, manufacturers always seek to find the best way to group parts with similar characteristics so that similar parts can be produced using specific processes in specific departments. Such problems are a common occurrence in industry, from manufacturing to service industry. For the purpose of this study, these problems are called *grouping problems*. Noteworthy, these problems are inherently difficult to solve because of their combinatorial nature [1] [2] [5]. However, they have similar grouping structures and characteristics upon which their solution approaches can be developed [5].

A study of grouping problems in the literature revealed interesting characteristics outlined as follows:

(i) they have a grouping structure that can be utilized in solution development;
(ii) they are highly combinatorial in nature, which makes them hard to solve;
(iii) they are highly constrained, which makes them complex;

Due to their complex nature, expert systems, heuristic and metaheuristic approaches have been used to solve various grouping problems.

Genetic algorithm (GA) is a potential approach for this purpose [3] [4]. GA is a meta-heuristic method based on the mechanics of copying strings according to their objective function values and swapping partial strings to generate successive solution spaces that improve over time. Its distinctive feature is the use of probabilistic genetic operators as tools to guide a search toward regions of the search space with likely improvement. Group Genetic Algorithm (GGA) is a modification of the conventional genetic algorithms originally developed by Falkenauer [3] for addressing grouping problems. Remarkable improvements and applications of the GGA are found in [6], [8], [19] [20], [21] and [27].

Given the widespread occurrences of grouping problems in industry, and the complex nature of the problem, it is essential to design a robust versatile tool that can solve the problems across disciplines, with little or no fine tuning.

In the next section, we identify typical grouping problems in industry. Section 3 presents a description of the group genetic algorithm. We map the grouping problems in Section 4, showing how the problems lend themselves to the algorithm. Section 5 presents concluding remarks and further research prospects.

2. Identifying Typical Grouping Problems

In this section we provide a taxonomic identification of common grouping problems in Industrial Engineering. Table 1 lists the grouping problems identified.

Table 1. Identified grouping problems in industry

No.	Grouping Problems	Selected References
1	Manufacturing systems	[3] [4] [5] [6] [7] [9] [10] [11] [12] [13]
3	Logistics operations	[14] [15]] [16] [17] [18] [19] [20] [21]
2	Healthcare operations	[8] [9] [10] [24] [28] [29]
4	Group Technology	[3] [4] [9]

2.1 Cell Formation in Manufacturing Systems

In a manufacturing system, the formation of machine cells is a major concern aimed at improving productivity [5] [6] [8]. This is achieved by grouping together machines that can operate on a product family that cause little or no inter-cell movement of the products.

Figure 1, for example, illustrates a manufacturing system comprising 3 cells: cell 1, 2, and 3, each consisting of groups of machines (m1,m5,m6), (m2,m3), and (m4,m7), respectively. Considering the process flows and the parts to be manufactured various manufacturing system configurations can be generated and evaluated using suitable metaheuristics such as GGA [8].

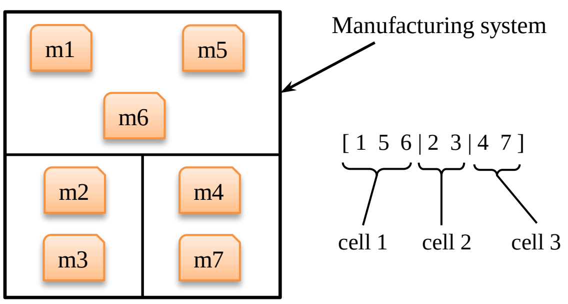

Figure 1. A typical manufacturing cell layout and its representation

2.2 Assembly Line Balancing

Line balancing is concerned with assignment of individual work elements or tasks to workstations with the objective of minimizing unit assembly cost [10] [12] [13]. Figure 2 provides a typical line balancing problem in which 7 tasks are to be allocated to 3 workstations. Groups of tasks (1,2), (3,4,5) and (6,7) are allocated to stations 1, 3, and 3 respectively. Howbeit, this candidate solution has to be evaluated to check if it provides the optimal cost. Further possible solutions may be generated and evaluated iteratively using suitable metaheuristic methods.

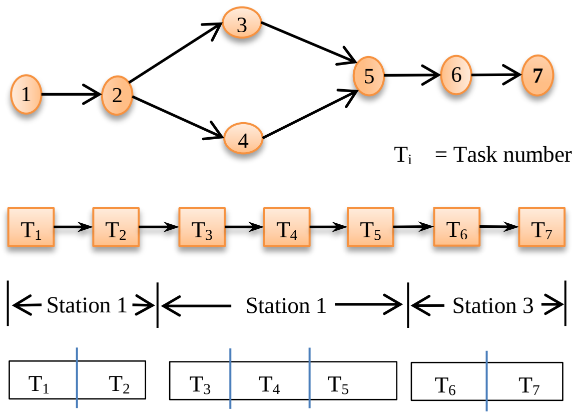

Figure 2. A typical line balancing problem

2.3 Logistics Operations

In logistics management, the vehicle routing problem (VRP) is a major challenge to decision makers. Optimizing the routing of vehicles is crucial for providing cost-effective services to customers [17] [18]. VRP is a hard combinatorial problem aimed at assigning groups of customers to a set of vehicles, such that the total costs incurred in visiting all the customers is minimized, subject to pertinent vehicle capacity, customer demand and time window constraints [19] [20].

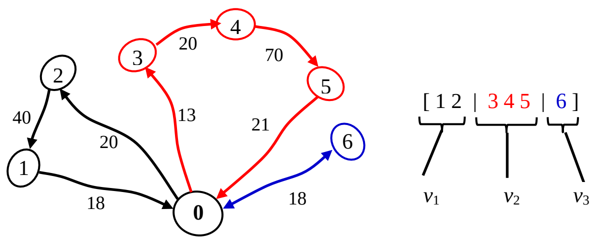

Figure 3. A typical vehicle routing schedule in logistics operations

Figure 3 shows a group representation of a typical VRP schedule comprising 6 customers that are assigned to 3 vehicles. Customer groups (1, 2), (3, 4, 5) and 6 are assigned to vehicles v_1, v_2, and v_3, respectively. The sequence of customers in each group signifies the order of customer visit or the route direction.

2.4 Home Healthcare Worker Scheduling

Typically, the home healthcare worker scheduling problem is described thus (see Figure 4) [21]: Consider a homecare center with m care givers to visit n clients, where each care giver k (k = 1,2,…,m) is supposed to serve patient j (j = 1,2…,n) within a given time window defined by earliest start and latest start times, e_j and l_j, respectively. The aim is to minimize costs of visiting clients [22]. If a care giver arrives at the client earlier than e_j or later than l_j, a penalty cost is incurred. Let a_j denote the time when a care giver reaches patient j, and p_e and p_l denote the respective unit penalty costs incurred when the care giver arrives too early or too late. Then, $\max[0,e_j - a_j]$ and $\max[0,a_j - l_j]$ have to be minimized, to maximize patient satisfaction. Furthermore, schedule quality should be maximized by constructing fair schedules within the limits of worker preferences [22].

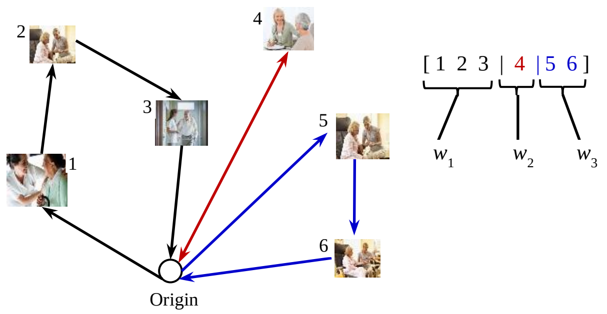

Figure 4. Homecare worker schedule example

2.5 Bin Packing Problems

The bin packing problem is a common hard problem in Industrial engineering where objects of different volumes must be packed into a finite number of bins or containers in a way that minimizes wasted space or number of bins used [23] [24]. For instance, in Figure 5, three bins, $b1$, $b2$, and $b3$ are packed with groups of objects (1,5), (3,4,2), and (6,7), respectively.

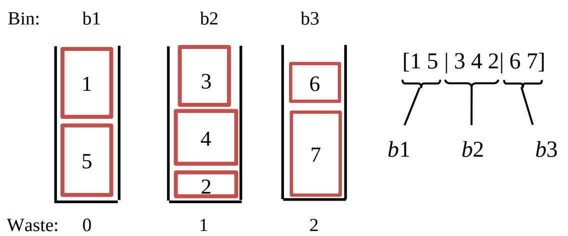

Figure 5. A typical bin-packing problem and its representation

There are many variations of the bin packing problem, such as 2-dimensional packing, linear packing, packing by weight, and packing by cost, with many applications, such as filling up containers, loading trucks with weight capacity constraints, metal cutting, and other related problems [25].

2.6 Task Assignment

The task assignment problem consists in assigning a set of tasks [27] [28], $T = \{1,...,n\}$ to an available set of workers $W = \{1,...,w\}$, where each task i is defined by duration p_i and time window $[e_i,l_i]$; e_i and l_i represent the respective earliest start and latest start times of the task [27]. Each worker has a scheduled working time of day. Oftentimes, it is required to limit the variation of individual workloads within acceptable limits [28]. Time window constraints should be satisfied. Figure 6 gives an example of an assignment of groups of 6 tasks to 3 workers.

Assignee	Tasks assigned
$w1$	1, 2
$w2$	3, 4, 5
$w3$	6, 7

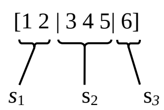

Figure 6. Example of task assignment and it's representation

2.7 Other Problems

Apart from the problems outlined in the previous section, other grouping problems exist in the industrial engineering field, such as districting problem and cutting stock problem, and other group technology applications [26].

The next section presents the general approach of the GGA approach.

3. Grouping Genetic Algorithm Approach

GGA is an improvement from genetic algorithm, aimed at taking advantage of the group structure of grouping problems. GGA's main elements are chromosome representation, population generation, fitness function evaluation, and the genetic operators (selection, crossover, mutation, inversion, and diversification). Figure 7 shows the basic logical flow of the GGA algorithm, together with its constituent operators.

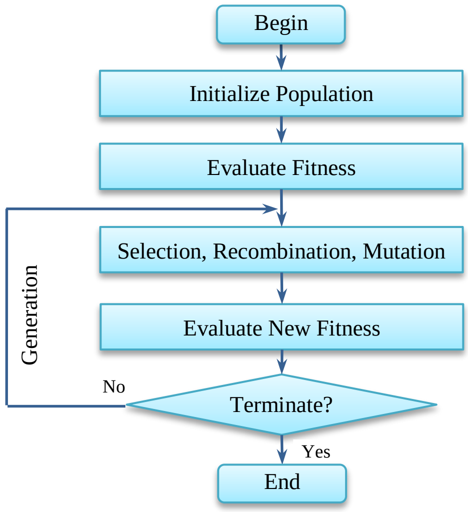

Figure 7. An outline of group genetic algorithm

3.1 GGA Coding

GGA approach begins by exploiting the group structure of a grouping problem, and developing a genetic code that represents a candidate solution (chromosome). The code comprises group of genes (digits) that show how items are grouped to form a candidate solution. Thus, in the *population initialization* phase, a number of chromosomes are randomly created and evaluated for goodness using a fitness function. The chromosomes are then fed into an iterative loop of genetic operators: selection, crossover, mutation, and inversion.

3.2 Selection

Selection involves mapping a cost function $g(s)$ of a chromosome s to a fitness function $f(s)$ for evaluation. The fitness function for each chromosome determines the chromosome with the maximum fitness value. The goal of optimization is to maximize $f(s)$, obtained thus (Goldberg (1989);

$$f(s)=\begin{cases} f_{\max}-g(s) & \text{if } g(s)<f_{\max} \\ 0 & \text{if otherwise} \end{cases} \qquad (3)$$

where, $g(s)$ is the cost function of the chromosome; and $f_{\max}$ is the largest cost function in the current population. By remainder stochastic sampling without replacement (Goldberg, 1989), each chromosome s is selected and stored in the *mating pool* according to its expected count e_s,

$$e_s=\frac{f_s}{(1/popsize)\sum_{s=1}^{popsize} f_s} \qquad (4)$$

where, f_s is the fitness function value of the s^{th} chromosome. Each chromosome receives copies equal to the integer part of e_s, that is, $[e_s]$. The fractional part of e_s, $fract(e_s)$, is treated as success probability of obtaining additional copies of the same chromosome into the mating pool or temporal population, called *temppop*. Chromosomes with higher fitness will have e_s, and higher chances of surviving into the next generation.

3.3 Crossover

Crossover is an evolutionary mechanism by which selected chromosomes mate to produce a pool of new offspring, called *selection pool*. Groups of genes are exchanged with probability p_c until the desired pool size *poolsize* = $popsize \times p_c$, is obtained. Figure 8 illustrates crossover operation.

Chromosomes:	New offspring:
[1 2 | 3 4 5 | 6]	[1 2 | **3 1** | 5 6]
swap	swapped
[2 4 | 3 1 | 5 6]	[2 4 | **3 4 5** | 6]

Figure 8. Crossover operation example

After crossover, some genes in the offspring may be redundant, while some may be missing. A repair mechanism is applied: identify and eliminate duplicated genes to the left of crossover point, and add missing genes. The group coding scheme takes advantage of the group structure. Figure 9 shows an example of repair mechanism [1 2 | 3 1 | 5 6].

Before repair	:	[1 2 | 3 1 | 5 6]
Eliminate 1	:	⇩
		[2 | 3 1 | 5 6]
Introduce 4	:	⇩
After repair	:	[2 4 | 3 1 | 5 6]

Figure 9. An example of the chromosome repair mechanism

3.4 Mutation

To intensify local search and to maintain population diversity, two types of mutation operators are applied to every new chromosome: *swap mutation* and *shift mutation*. Swap mutation exchanges genes between two randomly chosen groups in a chromosome. Figure 10 illustrates swap mutation; genes 2 and 3 are randomly chosen from trips 1 and 2, and swapped.

Chromosome	:	[2 4 | 3 1 | 5 6]
Select groups	:	2 and 3
Select and swap genes:	:	3 and 5
Mutated chromosome:	:	[2 4 | **5** 1 | **3** 6]

Figure 10. An example of swap mutation

The shift mutation operator randomly selects a frontier between two adjacent groups and shifts it by one step either to the right or to the left, as illustrated in Figure 11.

offspring	:	[2 4 | 3 1 | 5 6]	
select frontier, rand (1,2)	:	2	shift frontier
select direction	:	right	
mutated offspring	:	[2 4 | 3 1 5 | 6]	

Figure 11. Shift mutation example

3.5 Inversion

To prevent premature convergence, inversion rearranges, in reverse order, the groups of chosen chromosome, prior to crossover operation [2]. Figure 12 illustrates the inversion operation.

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

Figure 12. Inversion example

4. Mapping Grouping Problems to Grouping Genetic Algorithm

Grouping problems in industrial engineering can generally be represented by a common group structure that conveniently lends itself to the GGA approach [17]. Various examples have been given in Section 2. Based on this notion, we illustrate how the general group structure can be mapped or coded into the GGA approach: Consider a problem with 6 elements to be assigned to 3 assignees.

Figure 13 shows a coding scheme for grouping problems, comprising *code* 1 and *code* 2. We let code 1 represent a typical set of groups of elements, that is, (1,2), (3,4,5), and (6). The groups are separated by the symbol "|". These groups are assigned to the respective assignees A_1, A_2, and A_3. Furthermore, we let code 2 represent the respective positions of the delimiters or frontiers of the groups [20] [21] [27].

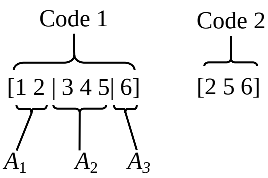

Figure 13. Mapping group problems using a group coding scheme

Clearly, it stands out that many, if not all, grouping problems can be represented in this form, with little or no adjustment. In grouping problems, the aim is to determine the membership of each element possible groups or sets such that the overall assignment maximizes the objective of that particular system. Hence the most important step is how to map or code a grouping problem based on the proposed group coding scheme. Having coded the problem, the general flow of the GGA approach is basically the same. The enhanced algorithm was originally developed by Mutingi and Mbohwa [2] and applied in a number of problem instances [2] [20] [21] [22] [27].

5. Concluding Remarks

Grouping problems are a common occurrence in industry. As such, it is needful to find a common view of the problems so that a useful tool can be designed to provide a solution approach to a wide range of problems across disciples. In this study we identified and presented a number of grouping problems, from various disciplines. We identified the common group structure of the problems and visualized how they can be mapped into a common code. The proposed common coding scheme is useful when solving such problems using the GGA approach. The GGA meta-heuristic has unique enhanced features, including the group chromosome scheme, group crossover, group mutation, and chromosome repair mechanism. The group operators enable the algorithm to reveal the group structure inherent in a problem set.

To the practicing manager, the grouping approach described in this study is handy as it offers a structured way of solving problems. The approach provides a simplified way of mapping specific problems into a common structure that can be solved by the GGA. Furthermore, the approach is widely applicable to a number of problem situations. Therefore, developing the GGA approach into a decision support tool can be an added advantage to the decision maker in the field of Industrial engineering.

The grouping problems identified in this study are not meant to be exhaustive. Further applications of the grouping approach can be identified across disciplines. We intend to explore more application areas, and to further improve the quality of the approach.

References

[1] Onwubolu, G.C. and Mutingi, M. (2001) A genetic algorithm approach to cellular manufacturing systems. Computers & Industrial Engineering, vol. 39, 125--144.

[2] Mutingi M. and Mbohwa C. (2012) Enhanced group genetic algorithm for the heterogeneous fixed fleet vehicle routing problem. *IEEE conference on Industrial Engineering and Engineering Management*, Hong Kong, December, 2012.

[3] Falkenauer E. (1992) The grouping genetic algorithms - widening the scope of the GAs. *Belgian Journal of Operations Research, Statistics and Computer Science*, vol. 33, pp. 79-102.

[4] Man, K. F., Tang, K. S. and Kwong, S. (1999) *Genetic Algorithms: Concepts and Design*, Springer, London.

[5] Filho E.V.G. and Tiberti A.J. (2006) A group genetic algorithm for the machine cell formation problem. *International Journal of Production Economics*, vol. 102, pp. 1-21.

[6] Mutingi M., Mbohwa C., Mhlanga S., Goriwondo, W. (2012) Integrated cellular manufacturing system design: An evolutionary algorithm approach. *Proceedings of the 3rd International Conference on Industrial Engineering and Operations Management*, Turkey, 3-6 July 2012; pp 254-264 (Best Paper Award).

[7] Mahdavi, I. and Mahadevan, B. (2008) CLASS: An algorithm for cellular manufacturing system and layout design using sequence data. *Robotics and Computer-Integrated Manufacturing*, vol. 24, 488-497.

[8] Mutingi M. and Onwubolu G.C. (2012) Integrated Cellular Manufacturing System Design and Layout Using Group Genetic Algorithms. In Faieza Abdul Aziz (Ed.), *Manufacturing System.* InTech-Open, USA, pp. 205-222, ISBN: 978-953-51-0530-5.

[9] Won, Y. and Lee, K. C. (2001). Group technology cell formation considering operation sequences and production volumes. *Internation Journal of Production Research*, vol. 39, 2755--2768.

[10] Scholl, A. (1999) *Balancing and Sequencing of Assembly Lines.*Physica-Verlag, Heidelberg.

[11] Becker, C. and Scholl, A. (2006) A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research,*168, 694-715

[12] Sabuncuoglu, I., Erel, E. and Tanyer, M. (2000). Assembly line balancing using genetic algorithms. *Journal of Intelligent Manufacturing*, 11(3) 295-310.

[13] Scholl, A. and Becker, C. (2006). State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, 168, 666-693

[14] Toth P. and Vigo D. (2002) The Vehicle Routing Problem. *SIAM Monograph on Discrete Mathematics and Applications*, Philadelphia, PA: SIAM.

[15] Braysey O. and Gendreau M. (2005) Vehicle routing problems with time windows, part I: Route Construction and Local Search Algorithms. *Transportation Science*, vol. 39(1), pp. 104–118.

[16] Liu, S., Huang, W. and Ma H. (2009) An effective genetic algorithm for the fleet size and mix vehicle routing problems. *Transportation Research Part E*, vol. 45, pp. 434-445.

[17] Mutingi M., Mbohwa C. 2012. Group genetic algorithm for the fleet size and mix vehicle routing problem. *Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management*, Hong Kong, 10-13 December 2012.

[18] Taillard, E.D. (1999) A heuristic column generation method for the heterogeneous fleet VRP, *RAIRO* 33, pp. 1–34.

[19] Gendreau, M. Laporte, G., Musaraganyi, C. and Taillard E.D. (1999) A tabu search heuristic for the heterogeneous fleet vehicle

routing problem. *Computers and Operations Research*, vol. 26, pp. 1153–1173.

[20] Mutingi M. (2013) Group Genetic Algorithms for Heterogeneous Vehicle Routing. In Antonio David Masegosa, Pablo J. Villacorta, Carlos Cruz-Corona, M.S. García-Cascales, María Teresa Lamata, José Luis Verdegay (Ed.), *Exploring Innovative and Successful Applications of Soft Computing*. IGI-Global, USA (forthcoming), ISBN: 978-1-4666-4785-5.

[21] Mutingi M. and Mbohwa C. (2013) Home healthcare worker scheduling: A group genetic algorithm approach. In S. I. Ao and Len Gelman and David WL Hukins and Andrew Hunter and A. M.(Eds.) Korsunsky. *Lecture Notes in Engineering and Computer Science: Proceedings of The World Congress on Engineering* 2013, U.K., 3-5 July, 2013, London, pp 721-725 - London, UK

[22] Mutingi M. and Mbohwa C. (2013) Home healthcare worker scheduling: A group genetic algorithm approach. *Proceedings of The World Congress on Engineering* 2013, U.K., 3-5 July, 2013, London, pp. 721-725 - London, UK

[23] Allen, S. D., Burke, E. K. and Kendall, G. (2011) A hybrid placement strategy for the three-dimensional strip packing problem. *European Journal of Operational Research*, 209(3), 219–227. doi:10.1016/j.ejor.2010.09.023.

[24] Pillay, N. (2012) A study of evolutionary algorithm selection hyper-heuristics for the one-dimensional binpacking problem. South African Computer Journal, 48, 31–40.

[25] Ramesh, R. (2001) A generic approach for nesting of 2-D parts in 2-D sheets using genetic and heuristic algorithms. *Computer-Aided Design*, 33(12), 879–891. doi:10.1016/S0010-4485(00)00112-3.

[26] Onwubolu, G.C. and Mutingi M. (2003) A genetic algorithm approach for the cutting stock problem. *Journal of Intelligent Manufacturing*, vol.14, pp 209-218.

[27] Mutingi M. and Mbohwa C. (2013) Task Assignment in Home Health Care: A Fuzzy Group Genetic Algorithm Approach. *The 25th Annual Conference of the Southern African Institute of Industrial Engineering 2013, Stellenbosch, South Africa, 9-11 July*.

[28] Bachouch, R.B., Liesp, A.G., Insa, L. and Hajri-Gabouj, S. (2010) An optimization model for task assignment in home healthcare. *IEEE Workshop on Health Care Management (WHCM)*, pp. 1-6, 2010.