Acta Cybernetica **00** (0000) 1–11.

Cohesion and Balance in a Human Resource Allocation Problem

László Illyés*

Abstract

Collaborative work appears between intelligent agents of different types. The problem discussed occurred when many construction workers were taken to Germany from Romania to work in construction projects. Managers have to make independent groups of workers from some categories, like carpenters, brick layers, etc. To discover their collaborative attitudes they use the scoring method, where every worker scores the others from different trades. The objectives are to form groups of workers with high compatibility value and to have a high compatibility value for the worst group, too. The problem becomes more interesting if software collaborative groups or specialized intelligent agents are involved. One has to prospect also the level of knowledge overlap between the trade groups of agents. This paper resumes to the problem of construction workers so as there is no overlap between the trades and the level of knowledge is not in the universe of discourse. We propose a Greedy and a genetic algorithm approach and we compare these methods.

Keywords: human resource allocation, genetic algorithm, minimax problem

1 Introduction

Our world proceeds toward team work. Nobody can perform good tasks alone on a long term. Nitchi in [1] considers the collaboration as an intelligent activity based on 3C (Communication, Coordination and Cooperation). Collaborative work in a group needs compatibility analysis. One can perform this with personality and aptitude tests, but also with the scoring method, where the group actors score each other. This paper proposes to take in consideration a real-life problem that came up when many construction workers were taken to Germany from Romania to work in construction projects [2]. A managerial problem was to distribute them in independently working groups. And the objectives were that those groups should be balanced and with high compatibility factors. The meaning of 'balanced' in this case is that the least fitted group should have a maximum value.

*Sapientia Hungarian University of Transylvania from Cluj-Napoca, Faculty of Business and Humanities, Department of Mathematics and Informatics, Miercurea Ciuc, E-mail: illyeslaszlo@sapientia.siculorum.ro

In this paper we take in consideration the next relaxations without a loss of generality: we use only the scoring method (the aptitudes are not tested), there is no overlap between trade workers (a carpenter does not perform bricklayer tasks), inside a trade there is no scoring.
We organize the remainder of the paper as follows. In section 2 we provide the mathematical background. Section 3 and 4 show the Greedy and the Genetic Algorithm approach. Experimental results are shown in section 5. Finally, we draw some conclusions to this paper.

2 The mathematical model

We consider N the total number of workers, n the cardinality of trades, m_i cardinality of workers in trade i, m_{min} minimal cardinality of m_i, c_{ij} the score given by person i to person j. We have $\sum_{i=1}^{n} m_i = N$, $c_{ij} = 1, \ldots, 5$. In the case when worker i is in a same trade with worker j we have $c_{ij} = 0$. Considering the scoring matrix c_{ij} we construct the symmetrical collaboration matrix $v_{ij} = v_{ji} = c_{ij} * c_{ji}$. We denote with W the total social value of the groups formed. The mathematical model obtained is:

$$\max\Big(\sum_{j=1}^{N}\sum_{i=1}^{N} v_{ij}x_{ij}\Big) \tag{1}$$

where $x_{ij} = 1$ in case of worker i is in a same group with worker j and $x_{ij} = 0$ otherwise.
The second objective function reflects the balance between the groups. This objective leads us to a max-min or min-max problem. In case we have the formed groups U_k with $k = 1, \ldots, m_{min}$, the second objective has the next form:

$$\max\left(\min_{k}\Big\{\sum_{i,j\in U_k} v_{ij}\Big\}\right) \tag{2}$$

For our testing purpose we choose an objective function that reflects both initial objectives. If we use the previous notations, the new mathematical model obtained will be:

$$\max\left(\Big(\sum_{j=1}^{N}\sum_{i=1}^{N} v_{ij}x_{ij}\Big) + \min_{k}\Big(\sum_{j,j\in U_k} v_{ij}\Big)\right) \tag{3}$$

The first part of the formula reflects the first objective, the second, the second objective.
The objective function can be extended into this more general form, where $\propto (m_{min})$ is a function of the minimal cardinality trade:

$$\max\left(\Big(\sum_{j=1}^{N}\sum_{i=1}^{N} v_{ij}x_{ij}\Big) + \propto (m_{min}) * \min_{k}\Big(\sum_{j,j\in U_k} v_{ij}\Big)\right) \tag{4}$$

We cannot reduce the whole problem to a minimax problem for many reasons. One of this appears when we can form more groups than needed.

3 The Greedy approach

Greedy algorithms, as the word suggests, mimic the behavior of the people, for example, in every moment they try to obtain the best result that the present situation offers. In this section we propose some Greedy approaches to solve the main problem or subproblems. We think that the Greedy coalition formation proposed by us is a very interesting one because it follows both objectives of the managers at the same time.

3.1 Greedy exclusion of workers

This procedure solves only a subproblem of the whole because after the exclusion of workers we have to try to proceed to the coalition formation. In case the workers cardinality in trades is not the same, one has to exclude some of the workers from the coalition formation. This is true even in cases when managers can form more groups than needed. The Greedy exclusion of workers can be applied to the score matrix c_{ij} or to the collaboration matrix v_{ij}. The Greedy exclusion using the score matrix means that we exclude the workers with the smallest sum of scores received from the others and who belong to trades with superior cardinality than m_{min}. The Greedy exclusion using the collaboration matrix means that we exclude from the same trades the workers with the smallest sum of collaborative values. We study the problem for a small numerical example having brute-force result to compare with the algorithm's results. Like we expected and in concordance with the defined objective functions, we find that the exclusion using the collaboration matrix performs better. We observe that both approaches can exclude the global optima.

3.2 Greedy coalition formation

This algorithm is considered to be in the focus of the paper. We consider the 'best individual' any worker who, included in the group, brings the highest collaboration value to that group. Algorithm 1 describes this approach. The Greedy approach for the first objective derives from Step 6 of the algorithm. The second objective is hidden in Steps 5 and 6. The worst group chooses first to gain some equilibrium.

When we construct the collaboration matrix we put the members of each trade in an adjacent position. This arrangement is useful for the implementation and understanding of the algorithm. As we see on the table presented, trades are {1,2,3,4}; {5,6,7,8} and {9,10,11}. Because of the last trade, we can form only 3 groups. In the first step, the algorithm gives the following three sub-groups: {9, 5}=20; {10, 2}=20 and {11, 6}=16. In the following step, the group with the

Algorithm 1 A Greedy-type algorithm

Funct GRW1

1: Choose one trade from the smallest cardinality trades.
2: Put each individual from that trade in a different group.
3: Initialize the group's collaboration values with 0.
4: **while** groups are not formed **do**
5: Put the coalitions in the ascending order of collaboration points gathered.
6: In the order obtained in the previous step, choose the 'best individual' from the remaining individuals belonging to the remaining groups.
7: Add the collaboration values between the new member and the old members to the collaboration value of the group.
8: **end while**

Table 1: One representation of a collaboration matrix

	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	2	12	10	6	4	5	3
2	0	0	0	0	6	3	5	5	2	20	1
3	0	0	0	0	15	25	1	3	2	6	3
4	0	0	0	0	5	2	8	15	4	12	6
5	2	6	15	*5*	0	0	0	0	20	3	3
6	12	3	*25*	2	0	0	0	0	15	5	16
7	10	*5*	1	8	0	0	0	0	12	12	5
8	6	5	3	15	0	0	0	0	12	5	8
9	4	2	2	*4*	*20*	15	12	12	0	0	0
10	5	*20*	6	12	3	5	*12*	5	0	0	0
11	3	1	*3*	6	3	*16*	5	8	0	0	0

smallest value will choose first. In the case of tie results, the algorithm chooses in a lexicographical order. The next step's results are the following groups: {11, 6, 3}=44; {9, 5, 4}=29 and {10, 2, 7}=37. In this case W=110 and the value of the objective function, named fitness function (that we will use in GA part) $F_f = W + V_{min} = 139$. The values used are emphasized in the matrix.

3.3 Greedy coalition formation when supply dominates

In this case we can form more groups than needed. The algorithm uses the previous algorithm, denoted with GRW1:

Algorithm 2 A Greedy-type algorithm with domination of supply

Funct GRWD

1: Solve the problem with GRW1.
2: Keep the best result
3: **while** cardinality of groups is greater than necessary **do**
4: Eliminate the worker from the smallest cardinality trade used at step 1 in GRW1 that is from the worst group.
5: Solve the reduced problem with GRW1.
6: Keep the best result so far.
7: **end while**

4 Genetic algorithm approach

A genetic algorithm (GA) is a heuristic method that mimics the evolution of species, described in [3] by Charles Darwin. Holland, in [4] describes the canonical genetic algorithm. In such an algorithm we use a population of individuals. Those are coded in binary strings (chromosomes) that describe the solution space of the problem. The first population is generated in a pseudo-randomized mode. This population is evolved by the GA in certain cycles, named generations. An accuracy value, named value of the fitness function gives us how 'good' the solution is. One can calculate the fitness value after decoding the chromosome. The best solution of the problem has the best fitness value too. This value is used in the reproduction process. The individuals with high fitness value have more chance to reproduce their genes. Standard implementation of the principle is the roulette wheel, fitness based selection. After the selection, the selected individuals are submitted to other genetic operators: crossover and mutation. The first operator combines two strings of chromosomes, the second modifies a single chromosome in a few bits. Important factors of efficiency of the GA are the probability values of applying these operators; the first contributes to the exploitation, the second to the exploration of the search space. De Jong in [5, 6, 7], gives some numbers for the 'optimal' parameters of the GA.

4.1 Chromosomes formed by independent gene chains

Permutation design of an individual (chromosome) is introduced to handle the TSP (Traveling Salesperson Problem). In this case we have special codification of chromosomes, every gene representing a specific city. The order of the cities in each chromosome represents the order of traveling, consequently genes are not substitutable, every gene has to be in all chromosomes of all generations. Because of this, we have specific genetic operators to maintain the structure of the chromosomes. We have crossover operators: PMX (Partially Mixed Crossover), OX (Order Crossover), CX (Cycle Crossover), ERC (Edge Recombination Crossover), Greedy Crossover and mutation operators, like the inversion operator or the classical operator of permutation mutation where two bits of the chromosome are switched.

The genetic representation of our problem space leads us to chains of permutation chromosomes that form the main chromosome. Every chain codes a trade of workers. Workers from the first trade are denoted with a_i. One chromosome could be the following:
$\{a_1, a_2, a_3, a_4, a_5; b_1, b_2, b_3, b_4, b_5; c_1, c_2, c_3, c_4\}$
The decoding of this chromosome means that the groups formed are $\{a_1, b_1, c_1\}$, $\{a_2, b_2, c_2\}$, $\{a_3, b_3, c_3\}$, $\{a_4, b_4, c_4\}$and a_5, b_5 are not used in the coalition formation. To roam the total search space we don't need to permute all the chains from the chromosome. A crossover of a chromosome will be in the following way: we fix one chain from the minimal cardinality trades (in our case c_i), the rest of the chains will be submitted to a classical permutation crossover operator separately. In case we use the PMX (Partially Mixed Crossover) with hot points in the 3 and 2 positions in the 1st and 2nd sub-chains with the next parents:
$P1 = \{a_1, a_2, a_3, \|, a_4, a_5; b_1, b_2, \|, b_3, b_4, b_5; c_1, c_2, c_3, c_4\}$
$P2 = \{a_5, a_2, a_1, \|, a_4, a_3; b_4, b_1, \|, b_3, b_5, b_2; c_1, c_2, c_3, c_4\}$ the offsprings are:
$O1 = \{a_1, a_2, a_3, \|, a_5, a_4; b_1, b_2, \|, b_4, b_5, b_3; c_1, c_2, c_3, c_4\}$ and
$O2 = \{a_5, a_2, a_1, \|, a_3, a_4; b_4, b_1, \|, b_2, b_3, b_5; c_1, c_2, c_3, c_4\}$
If we decode the 2 offsprings they give us the next coalitions:
$\{a_1, b_1, c_1\}$, $\{a_2, b_2, c_2\}$, $\{a_3, b_4, c_3\}$, $\{a_5, b_5, c_4\}$, a_4 and b_3 are excluded
$\{a_5, b_4, c_1\}$, $\{a_2, b_1, c_2\}$, $\{a_1, b_2, c_3\}$, $\{a_3, b_3, c_4\}$, a_4 and b_5 are excluded
We use the mutation operator over every sub-chain. The probability of mutation for the whole chromosome is between 0.1% and 1%. We assume that the algorithm choses for mutation from the first trade, workers from positions 2 and 4:
$O2 = \{a_5, a_2, a_1, a_3, a_4; b_4, b_1, b_2, b_3, b_5; c_1, c_2, c_3, c_4\}$
$O2' = \{a_5, a_3, a_1, a_2, a_4; b_4, b_1, b_2, b_3, b_5; c_1, c_2, c_3, c_4\}$

4.2 Chromosome representation with control genes

Control genes mimic real gene structure behavior [8]. Every chromosome is formed by a chain of binary control genes and the permutation chains, representing the trades. Control genes can activate or deactivate the functionality of the corresponding permutation chain. In this problem, if the control gene for a chain is '1', then crossover and mutation genetic operators are performed on that chain (or trade) and no operator is applied in the other case. The complex crossover operator has two parts, one for control genes and one for the permutation chain. Since every chromosome has a control gene structure, when we apply the crossover operator we randomly choose only one of the control genes from the two parents. In case one control gene is 'not connected' (0), the first offspring inherits the whole corresponding sub-chain from the first parent without modification, and the second offspring inherits in the same way from the second parent. The 'connected' (where control gene is 1) sub-chains are subject to permutation crossover operators. Those operators are applied to the same chains (trades) from the two parental chromosomes. One can use all binary genetic operators known to obtain the offspring's control genes. Because we don't want to have any more mutational effects for this combinatorial space, we use the one point crossover operator and a single-gene mutation

operator with small probability for the control genes.

Figure 1 shows an example of control gene structure. Trades formed by workers are $\{1, 2, 3, 4\}$; $\{5, 6, 7, 8, 9\}$; $\{10, 11, 12, 13\}$; $\{14, 15, 16, 17\}$. Trades 2 and 3 are 'connected' and subject to genetic operators.

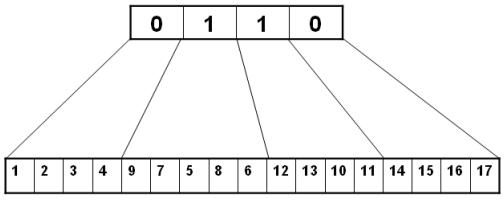

Figure 1: Example of control genes structure

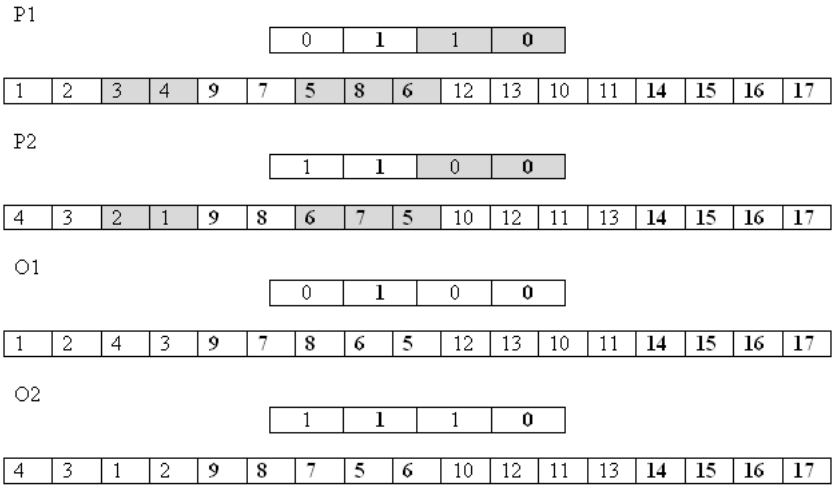

Figure 2: Complex crossover-PMX and single hot point

Figure 2 shows us how the complex crossover operator works: we have PMX operator for permutation chains when we use the second parent's control genes and single point crossover for the control chain. In this case the first and the second trades are subject to the crossover operator. We assume that the random generator chose the hot points between the 2nd and 3rd worker for both trades. The control genes are subject of a single point crossover operator with a hot point between the 2nd and 3rd genes. Trades are: $\{1, 2, 3, 4\}$, $\{5, 6, 7, 8, 9\}$, $\{10, 11, 12, 13\}$ and $\{14, 15, 16, 17\}$. The fitness function used for testing is defined in the Mathematical model section.

5 Comparative results of the methods

5.1 Implementation

The program is implemented in Java language. The genetic algorithm part is implemented using the Jgap3.3[9], freely available GA package. The whole structure of the Jgap and a reused code from A. Mescauscas, N. Rotstan, K. Meffert is used. The main reused code is from TSP problem solved by A. Mescauscas[10]. We implement the cycle crossover operator and the composed crossover operator for the control gene structure reusing the Greedy crossover operator code. We also implement the fitness function for this problem. We use a population with 512 chromosomes, a generation number of 128, crossover rate of 100% and mutation rate of 1%.

5.2 Results obtained from the three data sets

We save the two generated data sets in files and we use those in the testing process. We have 20 workers and 4 trades in the first set and 120 workers and 6 trades in the second set. The third data set is the student database collection, where students were asked about their preferences and the trades are formed in a random process. The students had known each other for 2 years. An introduction to this problem is treated in [11].
GA with independent gene chains gives a better result only in the case of the student database (18 students) and maybe because the distribution is far from uniform one. Table 2 shows the results obtained for the student database. The Greedy solution of 492, at the bottom of the table is singular because the algorithm always chooses the best worker in lexicographical order. The genetic algorithms run 40 times to obtain these results and give us the min, max and avg values presented in the two columns inside the table for each type of GA (with or without control genes). All

Table 2: The students example results

	Genetic algorithm	Control genes GA
min	477.00	484.00
max	500.00	525.00
avg	488.90	502.90
Greedy	492.00	492.00

the results obtained with control gene structure are better than those obtained with the simple GA (with independent gene chains), and the simple GA performs better than the Greedy solution.
In the other cases of 20 and 120 workers the Greedy algorithm performs better. In order to increase the results of these cases, we combine the two methods. We generate the first population in the neighborhood of the Greedy solution. In this

case 'neighborhood' means that the first population chromosomes are derived from the Greedy solution chromosome by switching one pair of workers from a trade. If we want to define a greater neighborhood we can switch two or more pairs of workers.
The results obtained for the data set with 20 workers in 4 trades using the Greedy method, the simple GA (with independent gene chains), the GA with control genes and the combined Greedy+GA method are shown in Figure 3. The same results are shown for the 120 workers data with workers in 6 trades in Figure 4. As we see in both figures, the results obtained in the following ascendant order are: simple GA, GA with control genes, Greedy result and the mixed algorithm, when the first population of the GA algorithm is generated in the neighborhood of the Greedy solution. Like in the students case the Greedy method give here one result for each data set too. We note that if the workers cardinality increases, than the difference between the GA solutions with or without control genes are insignificant. The average values of both data sets results are better when we use the control-gene structure instead of independent gene chains. We can affirm that the algorithm with control-gene structure performs better than the simple GA with independent gene chains.

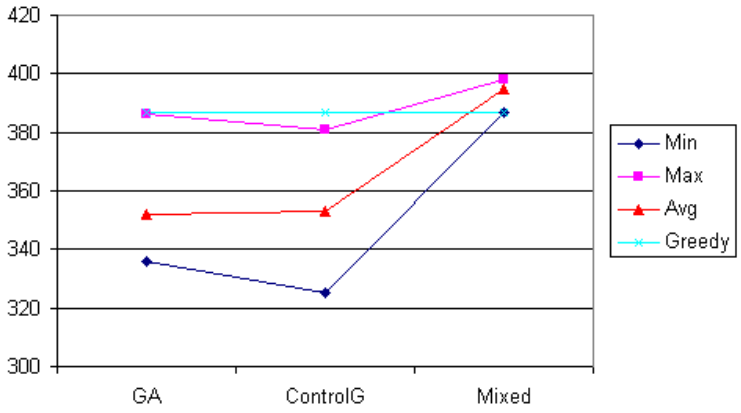

Figure 3: Results of the algorithms with 20 workers

6 Conclusions and future work

We solve this human resource allocation problem with a proper Greedy algorithm and with two genetic algorithms with different structures. The Greedy+GA algorithm depending on the cardinality of the workers has similar or better result than the Greedy result.
As future work we propose to search the students' preferences every year and analyze how to form better teams and what kind of networks they form. The knowledge

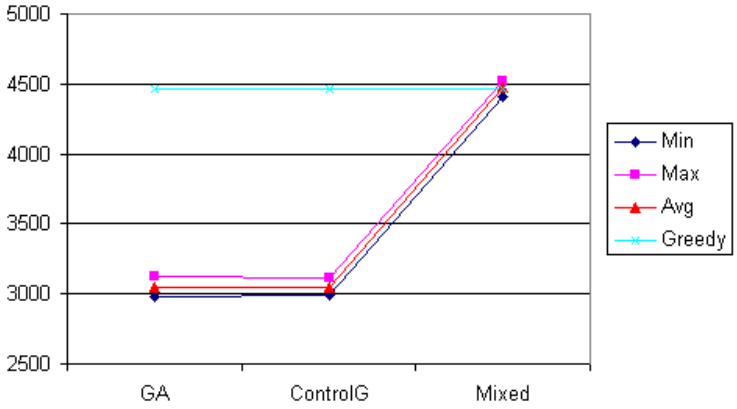

Figure 4: Results of the algorithms with 120 workers

of the students and some personality aptitudes are considered. Game-theory considerations are also a possible research topic for this problem, where we have an authority (the manager) and N players with different preferences. The players could lie if they are threatened to be eliminated. We can define types of individuals from the scores they give others if they do not lie. If somebody gives only high marks to the others, he may be exasperated about his job and hope to be chosen in a group. However, this idea would take us in the direction of psychology.

References

[1] Nichi . I., Avram-Nichi R. *On the Paradigm of Collaborative Support Systems*, In Proceedings of the Collaborative Support Systems in Business and Education, Academy of Economic Studies, Cluj Napoca, pages 274-292, 2005.

[2] Fábián, Cs. B. Private conversations 2003-2006.

[3] Darwin, C. *On the Origin of Species* John Murray, London, 1859.

[4] Holland, J.H. *Adaptation in Natural and Artificial Systems* Ann Arbour, University of Michigan Press, 1975.

[5] Jong, K.A.D. *An analysis of the behavior of a class of genetic adaptive adaptive systems.* PhD thesis, University of Michigan, 1975.

[6] Jong, K.A.D. A genetic-based global function optimization technique Technical Report, No.80-2, University of Pittsburgh, 1980.

[7] Jong, K.A.D. On using genetic algorithms to search program spaces. In Proceedings of the 2nd International Conference on *Genetic Algorithms and their Applications*, pages 210-216, Hillsdale, NJ, 1987.

[8] Hanli Wang et. al. Multi-objective hierarchical genetic algorithm for interpretable fuzzy rule-based knowledge extraction. *Fuzzy sets and systems*, 149:149–186, 2005.

[9] K. Meffert, N. Rotstan, Java Genetic Algorithms Package, http://jgap.sourceforge.net/

[10] A. Mescauscas, The traveling salesman problem http://jgap.sourceforge.net/doc/salesman/TravellingSalesman.html

[11] Illyés, L., Balanced student groups forming for university projects. In Proceedings of the 8th International Conference on *Informatics in Economy*, pages 554-559, Academy of Economic Studies, Bucharest, 2007

Received ...