ZESZYT NAUKOWY
Wyższa Szkoła Zarządzania i Bankowości w Krakowie

Dr Tadeusz Dyduch
The School of Banking and Management in Krakow
tdyduch@agh.edu.pl

ALGORITHM TO OPTIMIZE THE ASSIGNMENT OF PARTICIPANTS TO SEPARATE GROUPS WITH THE CONSIDERATION OF THEIR PREFERENCES

Introduction

The article presents the definition of the title issue, formalizes it and presents the selection and characteristics of the practically existing algorithms that provide the solution to the problem. It also analyzes the expressiveness of the means to express the preferences and restraints concerning the assignment participants.

The issue discussed here is well known practically; however, it has not been the subject of scientific definitions although it resembles Glover-Klingman production and distribution issues [1]. The article discusses the issue of Separate Group Assignment (SGA) with the consideration of constraints related to the numerousness of the groups and the preferences and restraints of the participants. It will also present the issue of multidimensional SGA, e.g. the assignment considering varied participant responsibilities within one schedule and with the exclusion of overlapping periods.

1. Definition of the canonical SGA problem (Separate Groups Assignment)

Determination of the distribution by n+1 subsets P_i for a given finite set of participants $\{u_i\} \in P$ at given constraints as regards the number of subsets.

$$|P| = m;\ \ P = \bigcup_{i=1..n} P_i \cup P_0;\ \ P_i \cap P_j = \emptyset\ \ for\ i \neq j\ ;\ \ |P_i| \leq q_i\ for\ i = 1..n$$

The distribution is to consider constraints (exclusions) regarding the possibilities to assign participants to particular subsets given by binary exclusion matrix $\{u_i\} \in P$ and to maximize the sum of participant preferences given by matrix *w[n;m]*.

$$\sum_i w(i,j) \rightarrow max;\ \ i = 1..n\ ^\wedge\ u_i \in P_j\ ^\wedge\ excl(i,j) = 1$$

With the assumption of the finiteness of the sets, it can be assumed without the loss of generalizability that:

- Preference matrix *w(i,j)* is non-negative: all values of *w(i,j)* can be always increased by a constant value without the change of the position of maximum value;
- The terms maximization and minimization of the preference function can be used interchangeably: the maximization of the sum of distinguished elements *w(i,j)* is equivalent to the minimization of the sum of elements $\overline{w}(i,j) = w_{max} - w(i,j)$

Finally, set P_0, which is hardly visible, should be mentioned. It is not restricted as regards its numerousness and it lacks the assignment of elements *excl()* and *w()*. This is a container for participants that cannot be assigned (or we are not capable of doing it ☺) to any other subset P_i at given conditions.

Moreover, it should be emphasized that it is acceptable that any of the named subsets of the distribution that constitutes the solution may remain empty.

2. Formalization of flow network

In the Glover-Klingman problem [1], which was mentioned in the introduction to the article, a flow network model was applied i.e. a directed graph *G(V,E,q)*, a unigraph with edge weights expressing flow capacity, with a distinguished pair of vertexes: source *X* and outflow *Y*. The well-known MaxFlow algorithms make it possible to determine flow function *fopt(i,j)* which provides the optimal flow value for each network edge e_{ij}. The values do not exceed the flow capacity of the edge, they maintain a zero balance of the flows in all indirect nodes and maximize total flow *F* from *X* to *Y*. $F \stackrel{\text{def}}{=} \sum_i f(i,Y) = -\sum_i f(X,i)$

If cost function $w(e)$ is given on flow network edges *G(V,E,q)*, the issue of the cost minimization of maximum flow MinCostMaxFlow can be presented.

The SGA problem can be described by the following flow network (Fig.1)

Directed graph *G (V ,E, w, q)* is given , where:

V – set of vertices v_i ,

E – set of edges e_{ij} ,

$w : E \rightarrow \mathbb{R}{\geq}0$ cost function given on edges,

$q : E \rightarrow \mathbb{R}{\geq}0$ flow function on edges h,

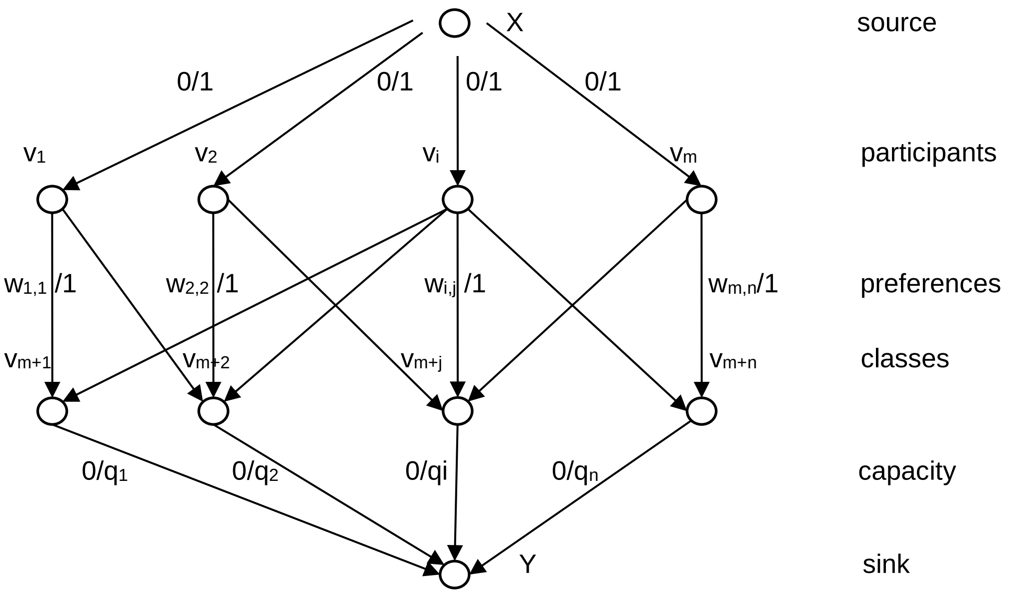
X , *Y* – distinguished pair of vertices, sources and outflow, respectively.

Function $f : E \rightarrow \mathbb{R}{\geq}0$ is referred to as flow which the following conditions:

1). $0 \leq f(e) \leq q(e),\ \forall\ e \in \boldsymbol{E}$ - edge-flow constraints

2). $\sum_{e_{i,j}} f(e_{i,j}) + \sum_{e_{j,i}} f(e_{j,i}) = 0, \forall v_j \in \boldsymbol{V} \setminus \{X, Y\}$ – condition for zero balance of flow in indirect nodes .

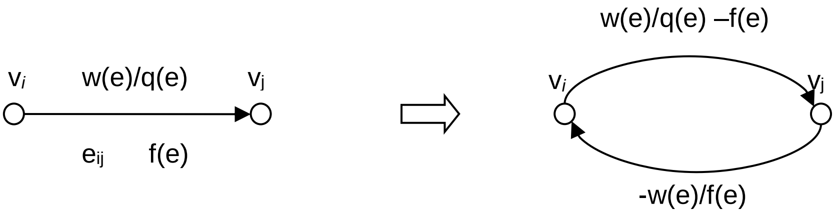
Fig.1. Diagram of flow network for SGA



If the total cost is the value of $\sum_{e\in E} w(e)\, f(e) = Q(f)$, then the issue of optimization will be expressed by the determination of the maximum flow that minimizes cost $Q(f)$: $Q(f) \rightarrow min$.

Among the well-known algorithm for the MinCostMaxFlow problem, the successive shortest path algorithm, Buxacker-Goven [2] is the easiest to explain. It applies the notion of residual flow network $G_{res.}$ This is a construct that describes the capability to manage the network (flow change area) with the existing non-zero flow f . For a given flow network G with a non-zero flow f, the residual network G_{res} is the network in which every edge e_{ij} with flow $q(e)$, cost $w(e)$ and flow $f(e)$ >0 is replaced by a pair of oppositely directed edges where e_{ij}' retains the remaining flow capacity $r(e')=q(e)-f(e)$ and cost $w(e')=w(e)$, while the opposite e_{ji} has flow capacity $r(e_{ji})=f(e)$, and cost $-w(e)$, as presented in Fig.2.

Fig. 1



The successive shortest path algorithm [2] is as follows:

∂=1; G_{res}=G

while ∂>0 repeat:

{ ∂=0;

}

- Find the cheapest connections d(j) from node X to the other nodes of residual network G_{res} (the current one) with the use of, e.g. Dijkstra algorithm:
- for $(e_{ij} \in E)\ w_e' = w_e - (d(j) - d(i))$;
- If there is P – the cheapest path from X to Y in G_{res};
- $\partial = min\{r_{ij}: (i, j) \in P\}$;
- increase flows on path P by ∂ ;
- update network G_{res} ;

}

The solution is represented by final flow F given by a set of edges with negative costs of final residual graph G_{res} .

Finally, let's consider an insignificant, nontrivial example [7], which will illustrate threats that may be faced by an inaccurate algorithm. There are10 applicants for classes organized in two groups with 3 places in every group. A competency test was conducted and declarations were collected regarding the preferences for the participation in particular groups. The matrix of data is given below. The solution has been distinguished. The algorithm, which initially will assign applicant 2 to group I, must have the option to change the assignment. Otherwise, applicant 9 with a much lower rank, will be accepted instead of candidate 5.

Fig. 2. Illustration of the problem solved.

Rank	10	9	9	8	8	7	7	5	4	3
I term	1	1	0	1	1	1	1	1	0	1
II term	0	1	1	0	0	0	0	1	1	1

3. Application range of the flow network model in SGA

The accepted algorithm ensures a maximum flow, i.e. the highest number of participants assigned to the groups. It allows for the possibility to leave a participant without the assignment and to leave an empty place in the group. There are no constraints in the number of participants with regard to the number of places in the groups. Moreover, the set of exclusions does not have practically any restrictions. However, the preferences and the information whether to maximize or minimize their sum should be given by numerical matrix W[m,n].

As a result, it is possible to assign applicants by their grades or ranks (see Fig.2), and to express their preferences by ranking their choices, by distributing between them a fixed number of weights or assigning to them any limited values. In the latter case, the participant – by giving more weights - increases his/her chances for assignment. This property of the MinCostMaxFlow algorithm – i.e. the availability of various weights to express preferences - will be used to solve multidimensional SGA.

3. Multidimensional SGA

By Multidimensional SGA (MSGA) the author understands a simultaneous determination of solutions for several fixed SGAs with the possibility of mutual exclusion of some groups from different SGAs. This requires the adjustment of several SGAs to avoid collisions.

In such cases, it is a common practice to assign a limited number of weights to every participant to express his/her preferences in all choices made, including all SGAs.

Let's take a set of several SGAs $MSGA = \{SGA^i\}, i = 1..t$ and determine the conflict area of the groups (regarding time periods). To be distinguished, the objects of particular SGA^i will be additionally denoted by a superscript. Thus, the conflict area $Z_{cnf} = \{P_u^i, P_v^j\} \wedge i \neq j$ is a set of distinguished pairs of named subsets from different distributions (SGA problems).

Let's assume a set of the solutions to subproblems MSGA. $M\widehat{ASGA} = \{\widehat{SGA}^i\}\ i = 1..t$. The set will be contradictory (unacceptable) as long as there is participant p_k

$$\exists(p_k, \widehat{P_u^i}, \widehat{P_v^j}) : p_k \in \widehat{P_u^i} \wedge p_k \in \widehat{P_v^j} \wedge \{P_u^i, P_v^j\} \in Z_{cnf}$$

The search for the acceptable solution of problem MSGA consists in:

- the modification of the preference function of the participant that is in collision (conducted by the system);
- the modification by the matrix system of the participant's exclusions through the addition of exclusions that eliminate collisions.

The search for the solution can be conducted with the application of the evolutionary optimization algorithm that is described by Dyduch [3], [4]. The algorithm consists in a random, based on the evolution of species, search for some elements (parameters) of the solution and assumes the application of a local optimization algorithm to determine the remaining elements that constitute a complete solution to the problem.

5. Evolutionary Optimization Algorithm EO

The EO algorithm, which is described below by a Data Flow Diagram DFD (Fig.1) is a two-level iterative procedure that applies an optimization procedure on the lower level. It may be the simplex algorithm [6], flow network optimization or many other [5]. In Fig.4 they are represented by module 2.OPT. The rest of the diagram describes the evolution algorithm of the upper level. The algorithm searches randomly the values of variable elements (parameters) for which the optimization algorithm of lower level can determine the rest of the optimal solution. The subset of variables that is searched randomly is referred to as genotype. Variables whose values are determined by lower-level algorithm are referred to as phenotype. The evolution process consists in the introduction of changes (mutations) that are selected randomly into the genotype, with suitably defined probability distributions.

It is crucial that local minimums should be avoided (setbacks in the optimum search). Thus, a sufficiently diversified parental population is maintained and crossover is applied as a substantial change of genotype, while in smaller systems the selection of the initial genotype is recurred repeatedly.

It is also significant that the areas once searched are not searched again. Here the taboo search is applied which requires the saving of the list of solutions that have been checked.

The following notions that are necessary to understand EO algorithm should be defined.

Let u^i, v^i denote the values of vectors u, v determined in the i^{th} iteration of computations. Vector u^i is determined on the upper level, while vector v^i on the lower one. Pair (u^i, v^i) is referred to as subject i .

Genotype is vector u^i - temporary point in the optimization subspace that is determined on the EO upper level with the application of the random steering method by the evolution algorithm.

Phenotype is vector v^i - temporary point in the optimization subspace that is determined by the EO lower level deterministic evolution algorithm.

For a defined value of genotype u^i , the task to optimize the lower level can be given as follows. Determine v^i so that

$$Q^i = f(u^i, v^i) = min_{v \in V(u^i)} f(u^i, v)$$

where Q^i is the value of the quality criterion for subject (u^i, v^i).

The solution of the EO problem in the form of (sub)optimal point is obtained in the iterative procedure:

$$f(\hat{u}, \hat{v}) = min_{(u,v) \in U \times V} f(u, v) = lim_{i \to \infty} \left(min_{v \in V(u^i)} f(u^i, v) \right)$$

where $u^i = rnd(u^{i-1}, u^{i-2}, \ldots)$ and $rnd()$ is a random function whose probability distributions are subject to changes in subsequent iterations.

The EO algorithm is described in Data Flow Diagram DFD in Fig.4. It includes six main subprocesses and data storage ST. Continuous lines represent the transfer of objects and steering, while dashed lines represent steering.

The INI process generates initial points of evolutionary searches. There may be one or more genotypes, depending on the defined numerousness of the initial population. In the case of multimodal optimization, where local minimums occur, the INI process enables the generation of varied initial points.

The OPT process has two objectives. First, it solves the task of the lower-level optimization, i.e. it determines the values of vector v^i and quality function Q^i . Secondly, in the course of this process, by means of the postoptimal analysis auxiliary data w^i are determined. They are used in the modification of probability distributions during the MM process.

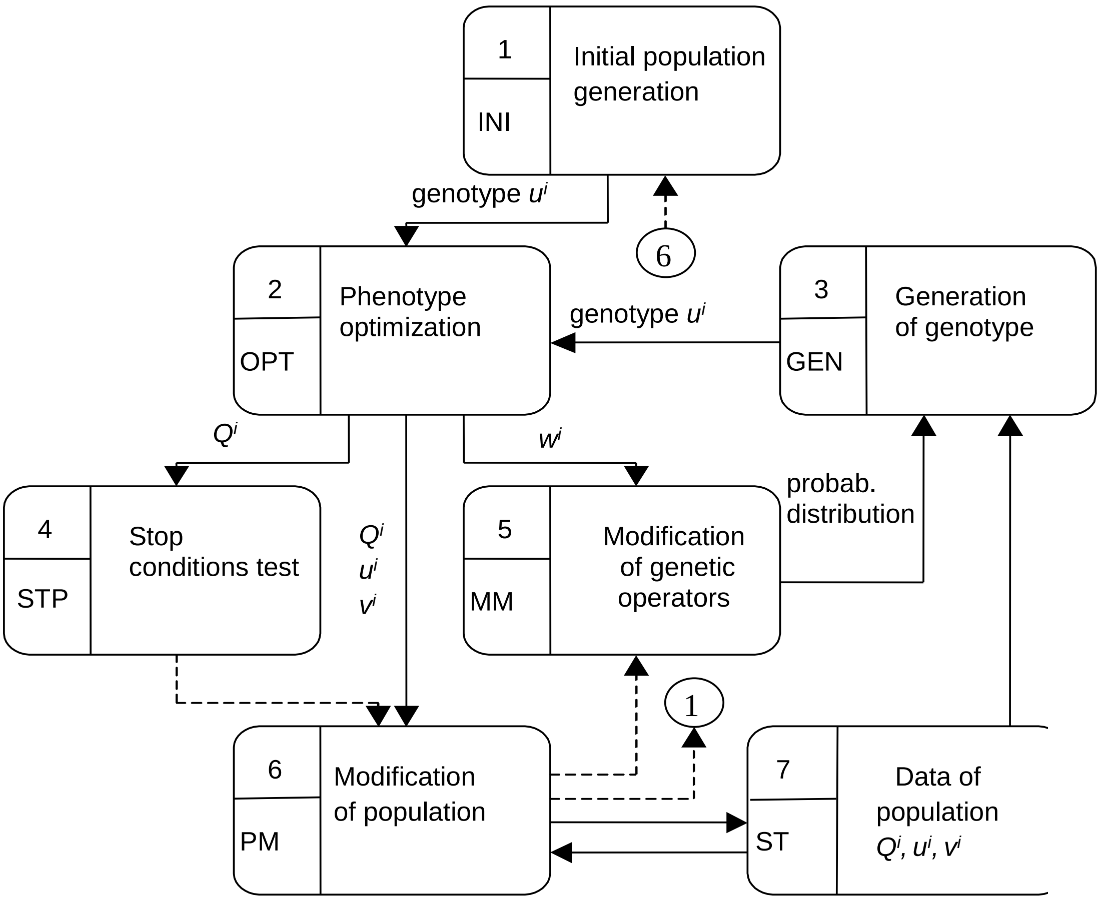
The GEN process generates new genotypes. It selects randomly the subject to be modified (a parent) from the current population stored in ST as well as the way of modification. Module MM provides the probability distributions that are applied by GEN. The generated subject is compared with the previously known ones that are stored in ST in order to avoid repetition (a taboo search).

The STP process checks the conditions for procedure stop. They include: the completion of the assumed computations from various initial points, the assumed number of iterations, the assumed number of iterations without any result improvement and other. When the stop conditions are not met, the computation process (steering) is transferred to module PM.

The MM process modifies the parameters of probability distributions that are applied in the GEN module to generate new genotypes. The modification of the parameters consists in the analysis of the so-far course of the iteration process and the accepted search strategy; it may also consider the results of the postoptimal analysis which provides the impact assessment of genotype variables on quality index $Q(u,v)$ in point (u^i,v^i). $w^i \cong \delta Q / \delta u^i$.

The PM process manages the number and quality of subjects that are stored in ST as a current population, in accordance to the accepted strategy. It also manages the list of the last subjects in ST,which is necessary in the taboo search. The principle is as follows: a new subject whose quality is better than that of the worst one in the population, can replace it on the condition that the population does not lose its variability. Moreover, if the quality index of the new subject is better than the index of the so far best one, the best known solution is consequently improved.

Fig 3. Data Flow Diagram of the Evolutionary Optimization Algorithm.



4. Application of EO algorithm to the MSGA problem

Let's denote optimal solutions for every separate problem by $\widehat{SGA}^i \stackrel{\text{def}}{=} \left(\widehat{Q^i}, \widehat{F^i}, \widehat{f^i}\right)$. The quality criterion of the MASG problem will be given by

$$\sum_{i=1..t} Q^i \left(max \sum_{i=1..t} F^i\right) \rightarrow max$$

We are looking for a maximum sum of participants' preferences in all SGA^i assignments when the condition of the maximization of the sum of all assigned places is met and -as defined in Chapter 5 - the principle of non-contradiction is applied. In this case, the use of the EO algorithm is as follows.

Phenotype, which is optimized by the MinCostMaxFlow algorithm, is a set of distributions given by functions $\{f^i\}$;

Genotype is a set of participant exclusion matrices in all SGAs.

The *OPT* algorithm here: the MinCostMaxFlow is applied to particular SGA^i whose order is selected randomly. After defining the $\widehat{SGA}^i$solution, for every case of the participant assignment to a subset that belongs to a pair of the conflict subsets

$$p_k \in \widehat{P_u^i} \wedge \{P_u^i, P_v^j\} \in Z_{cnf}$$

the exclusion matrix is modified in problem SPA^j with the exclusion of $p_k \in P_v^j$ and then consecutive SGAs are solved.

The *GEN* algorithm introduces changes to the part of the genotype that is responsible for conflicts. For each pair $\{P_u^i, P_v^j\} \in Z_{cnf}$ and every participant p_k whose exclusion function does not eliminate a simultaneous assignment to both subsets, the *GEN* algorithm may change bits (1,1) to (1,0) or (0,1) .

The *STP* algorithm is completed by testing the fulfilment of the condition of maximum assignment number. The value of $F_{lim} = \sum_{i=1..t} \hat{F}^i$ for $\hat{F}^i$ calculated for independently optimized SGAs constitutes a correct assessment of the maximum sum of the places assigned.

Conclusion

The aim of the article was to develop and implement creatively the accepted algorithms as well as to explain their implementative features and define their new application area, i.e. the multidimensional problem of the assignment to separate groups. The author hopes the article will inspire both operational researchers and software designers dealing with particular problems.

Bibliography

[1] Barr, R.S., Glover, F. & Klingman, D. *The alternating basis algorithm for assignment problems* Mathematical Programming (1977) 13

[2] Busacker R. G. Gowen P. J., *A procedure for determining a family of minimum cost network flow patterns.* Technical Paper 15, Operations Research Office, Johns Hopkins University, 1960.

[3] Dyduch T., *Adaptive Evolutionary Computation of the Parametric Optimization Problem*. Lecture Notes In Artificial Intelligence (3070), Springer-Verlag, Berlin , pp. 406-413, 2004.

[4] Dyduch T. Dudek-Dyduch E., *Two Level Adaptive Evolutionary Computation* Proc. of 23rd IASTED Int. Conf. Artificial Intelligence and Applications, Insbruck, Austria pp. 42-47, 2005

[5] Michalewicz Z., *Genetic Algorithms + Data Structures = Evolution Programs*,. Springer-Verlag, Berlin, 1996.

[6] Smith S., *The simplex method and evolutionary algorithms*. 5th Int. Conference on Evolutionary Computation ICEC'98, IEEE Press, 1998.

[7] www.matematyka.pl/400763.htm Problem przydziału -Badania operacyjne

Abstract

The article discusses the definition and formalization of the multidimensional problem of the assignment of participants to separate groups and presents the selection and characteristics of practically available algorithms that provide the solution to the problem. It also analyses the expressiveness of the means to present the preferences and constraints of the assignment participants.