BADANIA OPERACYJNE I DECYZJE

Nr 4 2009

Mariusz KALETA*,
Eugeniusz TOCZYŁOWSKI*

A COST ALLOCATION FRAMEWORK FOR **LP** AND **GLP** GAMES

Cost allocation problems within the GLPG class of games (**G**eneralized **L**inear **P**rogramming **G**ames) are considered in this paper. We assume that a group of agents participate in a common project and each agent defines his requirements for his expected benefit resulting from the project. Then, the joint cost of the project must be allocated amongst the agents in order to satisfy a set of required properties

Keywords: *cost allocation, cooperative games, linear programming game*

1. Introduction

The joint cost allocation problem appears in a situation where a group of agents participate in a common project, e.g. production of certain goods. Each agent defines his requirements for his expected benefit resulting from the project, which is usually called the agent's "demand". The total cost (joint cost) of the project depends on the agents' demands, that is, it is a function of these demands. We assume that this function is known. A nontrivial case appears if the joint cost is not equal to the sum of observed costs for each agent's demand considered separately. Then the following question arises: how should the joint cost be covered by the agents? Thus, the aim is to compute an appropriate allocation of joint costs to each individual agent. Throughout the paper we focus on the cost allocation problem, however this problem can be equivalently formulated as a profit share problem.

In practice, the cost allocation problem emerges e.g. in the joint production of goods [3], [14], [21], [25], [29], [31], [33], electricity (costs of grid usage, grid extension, generator start-ups, allocation of certificates) [22], [23], [34], [41], [42], network

* Warsaw University of Technology, Institute of Control and Computation Engineering, ul. Nowowiejska 15/19, 00-665 Warsaw, Poland, e-mail: M.Kaleta@elka.pw.edu.pl, E.Toczylowski@ia.pw.edu.pl

issues [5], [11], [12], [38], [39], [40], accounting [45], public utilities [1], [24], and other situations. Literature surveys can be found in [43], [46].

The main contribution of this article is the provision of a general and efficient algorithmic framework for computing exact cost allocations for a class of path-generated cost allocations in generalized linear programming games (GLPG).

Our approach is more general than the one closest to our work and given by SAMET et al. in [30]. Moreover, it does not benefit from any specific problem structure, in opposition to the algorithm given by SAMET et al. [30]. Our finding is new since previous articles considered specific problems, where the joint cost is usually sampled along some paths, which results in both – excessive computational efforts and approximation. For example, see fixed-cost allocations in power systems [41], allocation of firm-energy rights [6], production games, inventory games [4] and others.

2. Definition of the problem

Let us denote the set of agents by $\mathcal{M} = \{1, 2, ..., M\}$, where M is the number of agents, and denote the vector of the agents' real demands by $\underline{d} \in \Re^M$. The real demands are the requirements reported by agents that should be satisfied. Moreover, some other potential requirements, that agents could have submitted, will be considered internally during computations of the allocation. We call these the demands. A joint cost function $C : \Re^M \to \Re$ transforms the M-dimensional vector of the agents' demands into a real number – the joint cost of the project. The natural assumption is that there are no joint costs when there are no demands, that is, $C(0, ..., 0) = 0$.

For a given instance of a problem $(\underline{d}, C)$, the aim is to find a cost allocation $(x_1, ..., x_M)$ of value $C(\underline{d})$ to agents $1, ..., M$, where x_m is the cost allocated to agent m, $m \in \mathcal{M}$. However, usually we are interested in a more general approach of finding the cost allocation method (rule). Formally, the cost allocation method is a function $\phi(C, \underline{d}) = (x_1, ..., x_M) \in \Re^M$ which, for a given joint cost function C and demand vector $\underline{d}$, divides the joint cost $C(\underline{d})$ into individual costs x_m for each agent $m \in \mathcal{M}$. We also assume that the allocation must satisfy the budget-balance condition, that is, the sum of the values allocated to agents must be exactly equal to the joint cost: $\sum_{m \in \mathcal{M}} x_m = C(\underline{d})$.

Many versions of the cost allocation problem for joint projects have been extensively studied in the literature. A number of approaches have been proposed for some specific problems that are based on designing a procedure or mechanism – e.g. an arbitration rule, an auction or a competitive market mechanism – which produces the complete set of cost allocations [13]. Another possible approach to cost allocation is strictly normative, where game-theoretic methods of cost allocation play an important role [20], [46]. Cost allocation problems can be modeled as a cooperative game with transferable utility. Agents are considered to be players of such a game (we will use the terms "players" and "agents" interchangeably). Agents cooperate to perform a common project. However, each agent might completely or partially withdraw from the common project.

The agents' potential decisions form the feasible set of players' demands $\mathcal{D}$. Each configuration of demands from the space of feasible demands $\mathcal{D}$ forms a potential scenario of the agents' decisions for which the related potential joint cost is known or can be computed. We denote a single potential scenario by *d*. We also use the same notation for the real demands – $\underline{d}$.

In the simplest formulations, only discrete scenarios from $\left\langle 0;\underline{d}\right\rangle$ are considered during the computations. This means that agent *m* can only choose to participate in the common project at level $\underline{d}_m$ or not to participate. Such a discrete scenario *d* is called a *game coalition* (abbreviated to *coalition*). In this case, the demand space $\mathcal{D}$ consists of 2^M points.

In more general formulations of the cost allocation problem, agent *m* may choose the level of his participation to be anywhere between 0 and $\underline{d}_m$, so the space of scenarios $\mathcal{D} \equiv < 0;\underline{d} >$ is continuous. However, even in this case, many of the known allocation rules are based only on a subset of scenarios called coalitions.

Since the real decisions of the players are known and equal to $\underline{d}$, the game is somehow artificial. The agents' decisions are known, but not the rules of the game. Finding an allocation rule means finding the rules of the game such that the decision vector $\underline{d}$ is an equilibrium point of the game.

The most important game-theoretic allocation methods are the Shapley value [35], the core [10], [36], the nucleolus [32], the separable cost remaining benefits method (SCRB) [15], AUMANN–SHAPLEY pricing [2], and Ramsey pricing [28].

The *Shapley value* for agent *m* is a weighted average of the increase in costs after adding agent *m* to all the possible coalitions without *m*. To calculate the Shapley values, the costs for all coalitions must be computed (the number of coalitions is 2^M) – thus this approach could involve huge computational effort. The SCRB method is based on calculating the marginal costs for agent *m*. Initially, the SCRB method allocates to each agent *m* his marginal cost. Then, the remaining joint cost (positive or negative) not covered by the already allocated marginal costs is further allocated to agents in proportion to the so called remaining benefits, computed for each agent. The SCRB method considers coalitions of 1, $M - 1$ and *M* agents.

An important notion in cooperative game theory is the *core*, which specifies a set of solutions of primary importance. The *core* is a set of allocations such that there is no subcoalition, which can attain lower costs for its members than the grand coalition related to $\underline{d}$. However, in many practical situations the cost function *C* is usually non-concave (and not subadditive) and therefore the core is empty [36]. The *nucleolus* is a solution which is the "central" point in the core in some sense.

Other methods of cost allocation popular in the economics literature are AUMANN–SHAPLEY [2] and RAMSEY [28] pricing. Using the Aumann–Shapley pricing method, the cost x_m of agent *m* represents the average marginal cost along the ray from 0 to $\underline{d}$: $x_m = \int_0^{d} \left[\frac{\partial C(td)}{\partial d_m} \right] dt$. The function *C* must have continuous partial derivatives in the considered domain. Using the Ramsey pricing method, cost carriers are treated as new commodities and the cost of producing a bundle of commodities is allocated to each commodity. The demand for a bundle of goods depends on the allocated costs (and prices). In this case, the prices of all products are determined to maximize total consumer surplus over all the products minus the costs of production.

For some specific classes of problems certain cost allocation rules (that satisfy some desired properties of the allocation) have been proved to be the best. Usually some additional assumptions, like the subadditivity of the cost function *C*, are needed to achieve satisfactory allocation rules. There is still a number of open problems for which we do not know satisfactory allocation rules. We recall some examples in the following sections.

The framework formulated in this paper is appropriate for the allocation problem defined in this section under the assumption that it can be modeled as a non-atomic cooperative game with transferable utility (TU-game) with the joint cost function formulated as a linear programme. Within this framework, various allocation rules can be stated, meeting a given set of expectations related to allocation rules.

3. Path-generated allocations

A family of path-generated methods is a natural generalization of the Aumann-Shapley approach. Methods within this family compute the share of costs borne by agent *m* as the integral of the *m*-th partial derivative along a so called "path" in the demand space $\mathcal{D}$.

Definition 1. *For a given* $\underline{d}$*, a path function* γ *is a mapping* $\gamma : t \in [0,\infty] \times \mathcal{D} \mapsto \gamma(t) \in \mathcal{D}$*, where*

1. $\gamma(t)$ *is continuous and nondecreasing,*
2. $\gamma(0) = 0$,
3. *and there exists* $\hat{t} > 0$ *such that for all* $t \geq \hat{t}, \gamma(t) = \underline{d}$.

We denote the set of all paths by Γ. Any path from Γ links points 0 and $\underline{d}$ in $\mathcal{D}$.

Definition 2. *A path-generated allocation* $(x_1^\gamma, ..., x_M^\gamma)$ *imputed by path* γ *is defined by*

$$x_m^\gamma = \int_{t=0}^{\infty} \partial_m C(\gamma(t)) d\gamma_m(t) \tag{1}$$

Besides the Aumann–Shapley value, also the SHAPLEY–SHUBIK value [37] and serial cost sharing are examples of path-generated allocation rules. Path-generated methods have been formally studied by FRIEDMAN and MOULIN [7], [8]. The concept of such paths has simultaneously appeared in the works of Kaleta and Toczyłowski in the context of applications to electrical energy markets [16], [18]. They have considered a more general framework, which covers not only known paths like Aumann–Shapley and serial cost sharing, but also various paths based on local and global information [19].

4. Generalized linear programming games

The class of so called linear programming games (LPG) has been considered since the earliest works of SHAPLEY and SHUBIK [37], OWEN [27], and later work of GAMBLE and PAZGAL [9]. In such games the task of calculating the joint cost for a given demand scenario is formulated as a linear programming model with each constraint associated with one of the agents. Each agent defines a subset of the constraints in a linear programme, where this subset describes his requirements (agent demand balancing equations). One constraint of the linear programme is chosen by at most one agent.

Classically, this class of games is limited by two assumptions: 1) only two states are considered for each agent – their real or zero demands, 2) the corresponding variables in the agent demand balancing equations are set to 0 if an agent has demand equal to 0. Using this formulation, a scenario is defined by the coalition of agents together with their real demands.

In this paper we consider a generalized linear programming game (GLPG) by relaxing the above assumptions. We assume that the joint cost *C*(*d*) for an assumed demand scenario $d \in \mathcal{D} = 0, \underline{d} = [0, \underline{d}_1] \times [0, \underline{d}_1] \times ... \times [0, \underline{d}_M]$ can be calculated by solving the linear programming problem PP: $\{\min_{y,s} c^T y : Ay + s = [b; d]^T, y, s \geq 0\}$, where y represents the internal decisions related to the common project and s denotes surplus variables. The constraints can be divided into two subsets: the first subset is related to the logic of the problem, with right hand side b. The second subset $d = (d_m), m \in \mathcal{M}$ is related to the agents' demands from the set $\mathcal{D}$ and is indexed by the elements of $\mathcal{M}$.

One example of a GLPG – the transportation problem – has been studied by SAMET et al. [30]. In general, a cost function $C(d)$ based on linear programming is not differentiable. Even so, Samet et al. have shown that the structure of the problem allows us to calculate the Aumann–Shapley allocation. They provided an algorithm for determining the Aumann–Shapley allocation based on parametric programming for the transportation problem. However, this derivation uses the special structure of the transportation problem.

The model of the problem of balancing the *Single Commodity Infrastructure Market* (SCIM) is another important example of a game with a possibly empty core and non-subadditive characteristic function $C(d)$. This model is related to the problem of balancing an infrastructure market in which a group of suppliers and buyers compete in an organized manner. However, in opposition to pure exchange, free trade is limited by the infrastructure, e.g. a power grid, transportation network, or telecommunication network, where the feasible market balance is determined by a linear optimization problem. Some additional conditions must be included into the market balance, which appear in the model as additional constraints on resources. These additional infrastructure constraints affect the fairness conditions and make the economic benefits lower. The decrease in benefits resulting from infrastructure constraints is a joint cost of imperfect infrastructure. This joint cost can be allocated to each constraint on the infrastructure, which is essential for cost sharing by market players. Such allocation problems have become more and more important during the last decade, due to worldwide market liberalization and decentralization processes in many infrastructure sectors, including electrical energy, telecommunication, rail, water, urban transport and others. For example, balancing real-time electrical energy markets must take into account the fact that electrical power is transmitted along a transmission network according to Kirchhoff's laws. Sometimes the security of supply may require a minimum *must-run* level of power generation at a given node of the network. Hence, for a power supplier located at this node, some of his offers may have to be accepted irrespectively of the offered prices or costs. Acceptance of non-competitive offers may lead to the rejection of other, more competitive generators. Hence, the resulting additional cost of system usage must be allocated to the market players. As the SCIM game is non-convex, its core is usually empty. At present, no known allocation rule could be acknowledged as a satisfactory solution. There is still a need for further investigations into new path-based allocation methods. For a formal model of the SCIM problem and numerical example see [17].

5. Agents' demand space

We recall that for any $d \in \mathcal{D}$, the joint cost can be calculated by solving the linear programming model **PP**:

$$C(d) = \min_{y,s} c^T y \tag{2}$$

subject to:

$$Ay + s = [b; d], \tag{3}$$

$$y, s \geq 0, \tag{4}$$

where y is a vector of internal decisions related to the common project, s denotes surplus variables, b is the right hand side of the constraints modeling the problem's logic, and d is the vector of the agents' demands. Problem **DP**

$$C(d) = \max_{\pi, z} \pi^T [b; d] \tag{5}$$

subject to:

$$A^T \pi - z = c, \tag{6}$$

$$\pi, z \geq 0 \tag{7}$$

is the dual problem to (5)–(7), where the π are the dual variables in (6) and z denotes the vector of slack variables for the constraints in the dual space. Apart from the representation of the problem **PP** in the primal and dual spaces, parametric analysis can be defined over the agents' demand space $\mathcal{D}$. Any scenario $d \in \mathcal{D}$ defines the right hand side (RHS) of the linear programme **PP**. Movement in the space $d \in \mathcal{D}$ is equivalent to parametric analysis of the problem **PP**(d) under a perturbation of the right hand side. Below, we analyze some properties of the function $C(d)$ under perturbations of d.

Let us assume that for a given d we have obtained an optimal basic solution of the problem (5)–(7). Therefore, the linear programme can be represented in canonical form with the matrix $A = (B, N)$ partitioned into a basic matrix B and nonbasic matrix N. Similarly, the vectors of variables $y^T = (y_B^T, y_N^T)$ and parameters $c^T = (c_B^T, c_N^T)$ are partitioned into their basic and nonbasic components. The optimal value of the objective function, $C(d)$, equals

$$C = c_B^T B^{-1} [b; d], \tag{8}$$

where B is the basic matrix which is active for the optimal basic solution $y^T = (y_B^T, y_N^T)$ with $y_N = 0$ and

$$y_B = B^{-1}[b; d]. \tag{9}$$

In the space $\mathcal{D}$ the optimal value of the objective function $C(d), d \in \mathcal{D}$ is nonlinear, since the optimal canonical partition and the optimal base B are active only in some region and may change if d changes sufficiently. However, in some neighborhood of a given point d the optimal base is fixed and the formula (8) induces a hyperplane in the space $\Re^{M+1}$ of vectors (C, d). For certain boundary values of d, the optimal basis matrix B for the problem (5)–(7) may change and a new base induces another hyperplane in the space $\Re^{M+1}$. The intersection of these two hyperplanes projected into the space $\Re^{M}$ of parameters d forms a hyperplane that separates the adjacent regions in which these two bases are active.

In general, the space of RHS parameters $\mathcal{D} \subseteq \Re^{M}$ can be divided into polyhedral regions, convex closed polyhedral sets, which we call *cells* and denote by $\mathcal{C}$. In each cell $\mathcal{C}$ there is at least one active basis matrix B that satisfies (8), (9). The faces of a cell $\mathcal{C}$ overlap with some hyperplanes that separate the cell $\mathcal{C}$ from other cells – adjacent regions in which other bases are active. A cell in the parameter space is called *non-degenerate* if it contains an inner point.

Inside a non-degenerate cell it is possible to move in some neighborhood in any direction without changing the basis. Let us consider a degenerate cell which is the intersection of l, $2 \le l \le M$, adjacent non-degenerate cells. This degenerate cell has $M+1-1$ degrees of freedom. For the function $C(d)$ there are l different linear descriptions of the form (11). From the global point of view of joint costs, each description is equivalent, however each determines a different cost allocation for the individual constraints.

Let us consider the intersection of l, $2 \le l \le M+1$ adjacent non-degenerate cells and the corresponding bases $B_k, k = 1, ..., l$. If the intersection is nonempty, it constitutes a *degenerate* cell with $M+1-1$ degrees of freedom. Degenerate cells lie in certain subspaces of the parameter space, e.g. in faces, edges, or vertices. In a degenerate cell all the related bases are equivalent[1]. This is a case of primal degeneracy, where l – 1 basic variables are degenerate and equal to zero. The admissible points in a degenerate cell $\mathcal{C}$ are multiply equivalently defined and must satisfy the following l constraints

$$y_B = B_k^{-1}[b; d] \qquad k = 1,...,1 \tag{10}$$

where l – 1 degenerate basic variables must be equal to zero. Moreover, for any $[b; d] \in \mathcal{C}$ there are l multiple, equivalent definitions of the objective function

[1] Two bases are equivalent if the optimal values of the decision variables are identical, but the partitions of variables into basic and nonbasic are different.

$$C = \pi^k [b; d] \qquad k = 1, ..., l \tag{11}$$

with dual variables $\pi^k = c_{B_k}^T B_k^{-1}$.

The nonlinear function *C*(*d*) is given by the maximum from the set of hyperplanes that correspond to all the optimal bases locally active in various regions of the parameter space. Consequently, *C*(*d*) is a concave piecewise linear function – it is strictly linear in the domain in which one base is active. At the kinks of *C*(*d*), or equivalently at the points where the description of the function (8) changes, the function *C*(*d*) is given by the intersection of two or more hyperplanes.

The global parametric analysis of a linear programming problem can be represented in both the primal and dual spaces, but also in the RHS parameter space. All these representations are strongly related. In particular, local movement in the parameter space (inside a non-degenerate cell) leaves the optimal basis in the primal space unchanged, as well as the description of the objective function *C*(*d*) by the optimal dual variables, see (11). On the other hand, movement along degenerate cells, such as faces or edges, is related to multiple representations for many equivalent bases. Each representation (11) provides a different cost allocation for a given set of constraints.

6. Algorithmic framework for path-generated cost allocations

The algorithmic framework that we have developed is designed to compute the cost allocation for a GLPG problem for any path in $\mathcal{D}$. However, we will now show that any path in $\mathcal{D}$ can be transformed into an equivalent path built of linear segments. The following lemma holds:

Lemma 1. *Let d^i and d^{i+1} be in the same cell $\mathcal{C}$. Each subpath from d^i to d^{i+1} that belongs completely to the cell $\mathcal{C}$ determines the same cost allocation to each agent.*

Proof. The set of primal and dual solutions related to the cell $\mathcal{C}$ is constant. At each point of the cell the solution $(\hat{y}, \hat{s})$ remains the same, therefore the partial derivative π_m with respect to constraint *m* is the same. If we shift from point d^i to d^{i+1}, the shift in the *m*-th direction is equal to $\Delta d_m^i = d_m^{i+1} - d_m^i$. Generally, along any subpath from d_m^i to d_m^{i+1}, the *m*-th coordinate of the current point is shifted in one direction by $\delta^+ d_m^i$ and in the opposite direction by $\delta^- d_m^i$. It is obvious that $\delta^+ d_m^i - \delta^- d_m^i = \Delta d_m^i$.

Since the subpath belongs entirely to the cell *C*, the dual prices are constant and the allocation is equal to $\pi_m * \delta^+ d_m^i - \pi_m * \delta^- d_m^i = \pi_m * (\delta^+ d_m^i - \delta^- d_m^i) = \pi_m * \Delta d_m^i$. ■

Since each cell *C* is convex, from Lemma 1 we have the following:

Corollary 1. *Any subpath from d^i to d^{i+1} completely belonging to a cell gives the same cost allocation and*

1. Any such subpath can be replaced by a straight subpath – the shortest path between d^i and d^{i+1},

2. From the result described in point 1, one need consider only straight segments $(d^i; d^{i+1})$ of subpaths.

Additionally, the whole path linking points 0 and $\underline{d}$ can be transformed into a path built of I linear segments, where I is the number of different cells between 0 and $\underline{d}$. The local direction γ^i, which defines the shortest way from point d^i to d^{i+1} is related to segment $i \in 1, ..., I$. This path can be described by a set of local directions for all the cells crossed $\gamma = (\gamma^0, \gamma^1, ..., \gamma^I)$.

In consequence, we can divide the cost analysis into *I* iterations. In iteration *i* exactly one cell *C* is crossed. Fig. 1 shows a block diagram of the algorithm. First, the initial point $\underline{d}^0$ is set to 0, and the iteration counter is cleared. The next three blocks are related to crossing a single cell (one iteration). This procedure is repeated *I* times, until the end point $\underline{d}$ is reached. In each iteration the local direction γ^i is chosen for analysis. We will not focus on this block now, since it is related to a concrete allocation rule and needs to be specified the rule designer. In each iteration, two main steps are performed: 1) moving to the next cell along the given path; 2) computing the partial derivatives for the segment determined in step 1. When point $\underline{d}$ is reached, then the algorithm computes the allocation according to Definition 2.

Lemma 2. *Let us assume that consecutive moves in the space C performed by the algorithm depicted in Fig. 1 constitute a path linking points 0 and $\underline{d}$. The algorithm terminates after a finite number of iterations.*

The proof comes from the observation that any path, which is nondecreasing by definition, crosses a countable number of cells.

The dual problem DP to the PP task is as follows: $\max_{\pi,z} \pi^T [b; d]$: $A^T \pi - z = c, \pi, z \geq 0$. At the *i*-th iteration the current point $d^i \in \mathcal{D}$ and optimal solution of the pair PP^i and DP^i, denoted by $(\hat{y}^i; \hat{s}^i; \hat{\pi}^i; \hat{z}^i)$, are known, where PP^i and DP^i are defined at point d^i.

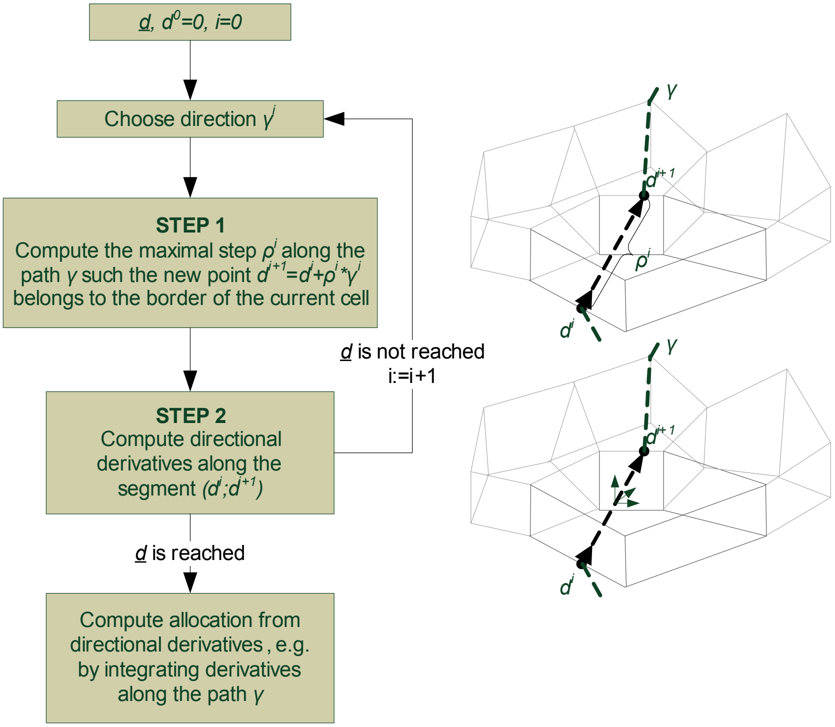
d, d⁰=0, i=0
Choose direction γⁱ
STEP 1
Compute the maximal step ρⁱ along the path γ such the new point dⁱ⁺¹=dⁱ+ρⁱ*γⁱ belongs to the border of the current cell
STEP 2
Compute directional derivatives along the segment (dⁱ;dⁱ⁺¹)
d is not reached
i:=i+1
d is reached
Compute allocation from directional derivatives , e.g. by integrating derivatives along the path γ
γ
dⁱ⁺¹
ρⁱ
dⁱ
γ
dⁱ⁺¹
dⁱ

Fig. 1. Block diagram of the algorithm

6.1. Step 1 – Reaching the opposite border of the current cell

At step 1, to compute the point on the border of the current cell reached by moving along the path γ^i, we use the notion of range of compatibility. At a given point d^i there may be a number of equivalent bases. At point d^i base *B* is compatible with direction γ^i (and γ^i is compatible with *B*), if *B* remains optimal for $d^i + \alpha^i \gamma^i$ for some $\alpha^i > 0$. The compatibility range of base *B* in direction γ^i is $\rho^i(B;\gamma^i) = \sup\{\alpha^i : B \text{ is optimal for } d^i + \alpha^i \gamma^i\}$. The range of the compatibility of base *B* can be determined from the canonical form of the linear programme by finding which is the first basic variable to achieve its upper or lower bound when α^i increases from 0.

Now, let us assume that we have fixed the direction γ^i at point d^i. The point d^i and direction γ^i define a cell in which each base compatible with γ^i is active. To determine the boundary point ρ^i of that cell, the optimization subproblem of finding $\rho(d^i, \gamma^i)$ must be solved. In the other words, we must find the maximum range of α^i in direction γ^i without leaving the current cell. Notice that the boundary of the cell is reached at a point induced by the minimum compatibility range $\rho^i(B; \gamma^i)$ of all bases B compatible with γ^i at point d^i. This range of compatibility $\rho^i(B; \gamma^i)$ can be calculated directly by solving the following problem $\mathrm{RC}^i(d^i, \gamma^i, \hat{\pi}^i, \hat{z}^i)$, where $\hat{\pi}^i$ and $\hat{z}^i$ give the solution of the dual problem at the point d^i.

Problem $\mathrm{RC}^i(d^i, \gamma^i, \hat{\pi}^i, \hat{z}^i)$:

$$\rho^i = \rho(d^i, \gamma^i) = \max_{y,s,\rho} \rho \tag{12}$$

subject to

$$Ay + s = [b; d^i + \rho\gamma^i]^T , \tag{13}$$

$$(y^i)^T \hat{z}^i = 0 , \tag{14}$$

$$(s^i)^T \hat{\pi}^i = 0 , \tag{15}$$

$$y^i, s^i \geq 0 . \tag{16}$$

The constraints (14) and (15) represent complementarity conditions. These conditions ensure that the boundary of the cell will not be crossed moving in direction γ^i. Increasing α beyond this boundary would result in a base change in the primal space, which would not be complementary to the fixed dual solution $\hat{\pi}^i, \hat{z}^i$. The new point d^{i+1} is equal to $d^i + \rho^i(d^i, \gamma)\gamma^i$.

6.2. Step 2 – computing the derivatives

Two consecutive points, let us say d^i, d^{i+1}, belong to a common cell. If both points d^i and d^{i+1} are in the same non-degenerate cell and at least one of them is in inside the cell, then the solution to the dual is unique along the path and the points inside the cell located on the path satisfy exactly one equation (11) for some k. The derivative $\partial^i_m C(\gamma(t; \overline{d}^i_m)) = \hat{\pi}^i_m$, which reflects the local unit cost, is constant along the segment i arises directly from (11) and is equal to π^i_m for the m-th agent.

Moreover, the additive properties of the costs associated with the constraints come from the form of (11).

If points d^i and d^{i+1} are in the same degenerate cell *C* which is the intersection of *l* non-degenerate cells, $2 \le l \le M+1$, then, while crossing the path from d^i to d^{i+1}, any one of the *M* +1 – *l* bases active in the cell can be chosen. For an arbitrary basis *k*, the equation (11) defines an arbitrary cost allocation of *C* according to the individuals' constraints, where the allocated costs sum up to the joint cost revealed by segment *i*.

Since in such a degenerate case there are *M* + 1 – *l* active bases related to the cell, we may consider *M* + 1 – *l* different allocations according to the dual prices appearing in (11). These allocations are not equivalent. Also, notice that if we choose an arbitrary base and allocate the costs according to this base, we shall allocate zero costs to the active constraints (for which the surplus variables are basic variables equal to 0, so their dual prices are 0).

If the move in the *m*-th direction immediately leads to entering another cell, then the derivative with respect to constraint *m* can be calculated as a one-sided derivative of the function *C*(*d*) in this direction. Therefore, the π_m^i can be determined from Mills theorem [26] as the maximal dual price of $\pi_m^{i,k}$ for all the bases *k* = 1, ..., *l* by solving the following:

Subproblem $\mathrm{DD}_i^m(\underline{\hat{y}}^i, \underline{\hat{s}}^i)$:

$$\hat{\pi}_m^i = \max_{\pi^i, z^i} \pi_m^i \tag{17}$$

subject to

$$A^T \pi^i + z^i = c\,, \tag{18}$$

$$(z^i)^T \underline{\hat{y}}^i = 0\,, \tag{19}$$

$$(\pi^i)^T \underline{\hat{s}}^i = 0\,, \tag{20}$$

$$\pi^i, z^i \ge 0\,. \tag{21}$$

where $\underline{\hat{y}}^i$ and $\underline{\hat{s}}^i$ give the primal solution at any interior point of the cell crossed in the *i*-th iteration, e.g. for the midpoint $d^i = 0.5(d^i + d^{i+1})$. The constraints (19) and (20) reflect the complementary conditions, which restrict the admissible solution space of problem DD to solutions corresponding to the current cell.

6.3. Integrating the local cost

After I iterations, the vector of steps $(\rho^0, ..., \rho^I)$ and matrix of derivatives $[\hat{\pi}_m^i]$ have been computed. The cost allocation can then be calculated directly from definition (1). However, the budget-balance condition will only be satisfied when the directional derivatives are additive, that is to say, when the following condition is satisfied:

$$\sum_{m \in \mathrm{M}} \hat{\pi}_m^i = \partial^i C(\gamma(t; d^i)) \tag{22}$$

for each segment $i \in 0, ..., I$, i.e. when the number of points from degenerate cells along the path γ is countable. The cost x_m^γ assigned to the m-th agent using path γ is calculated as the integral of the m-th partial derivative along the path γ:

$$x_m^\gamma = \sum_{i=0}^{I} \hat{\pi}_m^i \rho^i \gamma_m^i \ . \tag{23}$$

If condition (22) is not satisfied, then directly applying definition (1) will result in over-allocation – the sum of the allocated values will be greater than the total cost $C(\underline{d})$. To satisfy the budget-balance condition, the partial costs from segment i should be appropriately scaled, i.e. by a factor equal to $\frac{\hat{\pi}_m^i}{\sum_{m \in \mathrm{M}} \hat{\pi}_m^i} \partial_m C(\gamma(t), d^i)$.

The choice of the path in the agents' demand space is crucial to the quality of the method of allocation. The framework described above does not define such a path, which gives us some freedom to optimize the outcomes. Since any path can be used, this allows us to specify various cost allocation algorithms. In [19] we have described various groups of rules for choosing directions for the parametric analysis.

7. Numerical example

As a case study, we consider an example of an SCIM problem on the electrical energy market. All the individuals' demand constraints resulting from a given network of electrical energy generators lead to a joint cost, which has to be shared by each constraint. One type of such a constraint may be active during the winter, where the power generated in each power plant cannot be reduced below a minimum level. For instance, such constraints must be considered by the Polish Transmission System Operator (TSO) on the real-time balancing market due to security reasons. There are 20 generators and each of them submits 5 sale offers in staircase form. We assume that groups of generators are located within the network as given in Fig. 2. There are inequality constraints for the levels of power at the network nodes. We consider 10 constraints on the minimum level of generation and 15 constraints on the maximum level of generation. Demand is non-elastic and equal to 4500 MWh. To balance the market we use the SCIM model. The randomly generated sale offers are presented in Table 1. The values $y_n^{\max}, n = 1, ..., 5$ in the table are the maximum volumes for each offer 1, ..., 5 for a given generator and correspond to $p^{\max}$ in the original SCIM. The prices c_n for each offer correspond to *s* parameters in the SCIM model.

The total cost of balancing demand with the amount of energy produced in the unconstrained case with no demands from the agents is equal to $ 420 509. Under the given constraints, the total cost of producing this energy increases by $ 9 935 to $ 430 444. Thus the joint cost of all the resource constraints is $ 9 935.

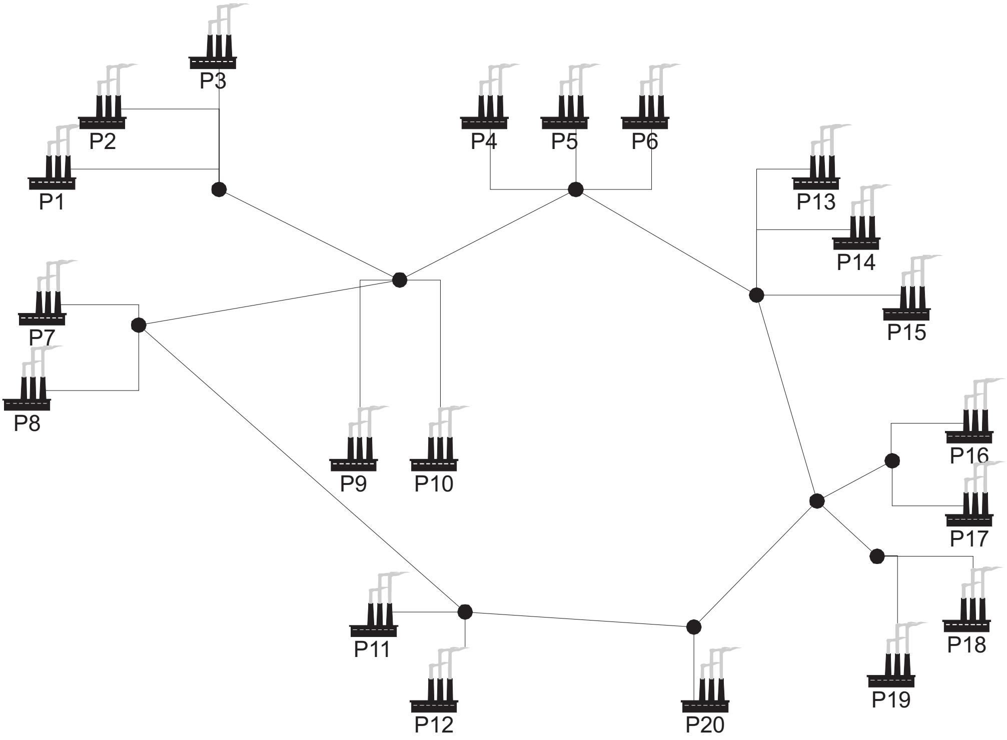

Fig. 2. Network topology

Table 1. Producers' offers (5 offers per generator)

Generator	$y_1^{\max}$	c_1	$y_2^{\max}$	c_2	$y_3^{\max}$	c_3	$y_4^{\max}$	c_4	$y_5^{\max}$	c_5
P1	120	82	40	115	40	132	40	133	40	146
P2	55	75	20	94	50	122	20	123	15	132
P3	200	110	30	120	30	125	60	132	30	148
P4	125	92	15	93	10	114	10	131	5	138
P5	120	73	40	87	50	92	90	132	50	140
P6	200	100	15	119	25	135	30	137	20	137
P7	110	109	30	115	40	125	40	128	10	148
P8	60	73	18	81	22	83	100	142	40	143
P9	150	76	100	77	50	86	25	93	25	147
P10	60	71	40	84	50	100	50	119	10	129
P11	70	72	50	86	50	114	50	137	30	147
P12	180	79	10	83	40	108	25	135	25	141
P13	150	87	20	94	70	98	40	128	30	144
P14	160	77	40	115	50	116	20	144	80	149
P15	190	91	30	115	30	123	25	128	5	139
P16	120	110	20	122	50	134	90	143	30	144
P17	130	76	35	95	35	96	75	134	55	149
P18	130	82	70	85	5	89	5	101	10	126
P19	160	72	10	76	20	111	50	126	110	149
P20	200	75	100	83	100	107	15	145	15	146

We use the parametric analysis described above with a path leading from the original RHS of the constrained problem to the first point of the relaxed problem where the joint cost is reduced to 0.

To set the local direction we applied the following rule: The local analysis direction γ^i at point d^i is to be set dynamically in each iteration of the algorithm as a vector with ones in the positions corresponding to *active* constraints and zeroes in the other positions. When entering a new cell at point d^i, the set of active constraints can be derived as follows. The constraint corresponding to demand *m* is considered active if and only if $\pi_m > 0$ for the problem DD^i. The new direction vector γ^i can be obtained by setting $\gamma_m^i = 1$ for the active constraints and setting $\gamma_m^i = 0$ for the remaining constraints.

Table 2 presents the results. For comparison, we calculated the decrease of joint costs in two cases: $\overline{x}_m$ is the decrease in costs after adding constraint *m* to the model without resource constraints, $\underline{x}_m$ is the decrease in costs after relaxing constraint *m* in the original model under all the resource constraints. x_m is the result of analysis in the RHS parameter space.

The first direction of analysis was determined by 7 active constraints and 14 non-active constraints. After 14 iterations this direction changed – other constraints were reflected in the local direction. After the next 8 iterations the algorithm terminated.

The algorithm applied using this rule for determining the direction of analysis allocates the costs mostly to active constraints (for which $\bar{x}_m \geq 0$) but also to some nonactive constraints which might be easily activated after the relaxation of other constraints. It can be observed that, in some cases, when a constraint is nonactive in the constrained model, but would disqualify any optimal solution for the unconstrained model ($\bar{x}_m \geq 0$), no costs are allocated. The reason for this is that it may be quite unlikely that such a constraint is activated when others are relaxed, although it is possible. In other cases the allocated costs vary between $\underline{x}_m$ and $\bar{x}_m$, which is a reasonable feature. The allocated costs are clearly related to the impact of the individual constraints on the joint cost of balancing.

Since we are not investigating the allocation rule here, but are testing our framework, we have chosen an example rule which explores the demand space extensively. The most important result from this case study is that our algorithm needs relatively few iterations (22) to find the allocations according to the chosen rule.

Table 2. Description of constraints and cost allocation

Constraint No.	Generators in constraints	Type	RHS	$\bar{x}_m$	$\underline{x}_m$	x_m
1	P1, P2, P3	≥	660	5	1365	1478.5
3	P7, P8	≥	320	0	2400	1716.7
7	P16, P17, P18, P19	≥	820	40	0	0.0
8	P20	≥	410	130	940	566.7
9	P16, P17	≥	600	2585	3060	3094.1
14	P9, P10	≤	520	95	0	0.0
16	P13, P14, P15	≤	735	305	125	155.3
20	P18, P19	≤	410	300	0	0.0
21	P7, P8, P9, P10	≤	700	1415	0	951.2
22	P1, P2, P3, P7, P8, P9, P10	≤	1330	650	1380	1218.0
24	P16, P17, P18, P19, P20	≤	1360	0	1265	754.5
25	P4, P5, P6, P9, P10	≤	1050	740	0	0.0

8. Summary

The GLPG class of cost games covers important cost allocation problems, including production games (based on the transportation problem), the Single Commodity Infrastructure Market game, and others. To the best of our knowledge, no general algorithm for the efficient computation of path-generated cost allocations for linear programming-based games (LPG and GLPG) has been developed before. In this paper we have formulated a procedure which is general in the meaning that it covers all possible paths. Thus it is applicable not only to well-known methods like Aumann–Shapley, but also to all potentially new path-based concepts. This procedure is based on the sensitivity analysis of a linear programme for determining the joint cost. However, the degeneracy and ambiguity of dual prices is a key problem during the cost analysis. According to the corollary formulated in this paper, each path of analysis has certain properties. We have applied these properties to formulate two optimization subproblems designed to: (i) compute consecutive steps along the path, and (ii) determine the local marginal costs. We have illustrated our approach using a numerical example of a transmission network cost allocation for electric power systems. Our results can be directly applied to examples of GPLG studied in the literature, where only approximate solutions are usually applied.

Bibliography

[1] AADLAND D.D., KOLPIN V., *Shared irrigation costs: en empirical and axiomatic analysis*, Mathematical Social Sciences, 1998, Vol. 35, No. 2, 203–218.

[2] AUMANN R.J, SHAPLEY L.S., *Values of Non-Atomic Games*, Princeton University Press, 1974.

[3] CHAMPSAUR P., *How to share the cost of a public good?* Int. J. Game Theory, 1975, Vol. 4, No.3, 113–129.

[4] CHEN X., ZHANG J., *A stochastic programming duality approach to inventory centralization games*, Operations Research, 2009, Vol. 57, No. 4.

[5] CLARKE D., SHENKER S., ZHANG L., *Supporting real-time applications in an integrated services packet network: Architecture and mechanism*, Comput. Comm. Rev., 1992, Vol. 22, No. 4, 14–26.

[6] FARIA E., BARROSO L. A., KELMAN R., GENVILLE S., PEREIRA M.V., ILIADIS N., *Allocation of firm-energy rights among hydro agents using cooperative game theory: an Aumann-Shapley approach*, Mini-EURO Conference, Operation Research Models and Methods in Energy Sector, 2006, University of Coimbra, Portugal.

[7] FRIEDMAN E., *Paths and consistency in additive cost sharing*, International Journal of Game Theory, 2004, Vol. 32, No. 4, 501–518.

[8] FRIEDMAN E, MOULIN H., *Three methods to share joint costs or surplus*, Journal of Economic Theory, 1999, Vol. 87, No. 2, 275–312.

[9] GAMBLE A.B., PAZGAL A.I., *A linear programming framework for network games*, Center for Mathematical Studies in Economics and Management Science, Discussion Paper No. 1119, 1995, Northwestern University.

[10] GILLIES D.B., *Some theorems on n-person games*, Ph.D. thesis, Department of Mathematics, Princeton University, 1953.

[11] HENRIET D., MOULIN H., *Traffic based cost allocation in a network*, RAND J. Econ., 1996, Vol. 27, No. 2, 332–345.

[12] HERZOG S., SHENKER S., ESTRIN D., *Sharing the cost of multicast trees: An axiomatic analysis*, IEEE ACM Transactions on Networking, 1997, Vol. 5, No. 6, 847–860.

[13] HOGAN W.W., *Contract networks for electricity power transmission*, Journal of Regulatory Economics, 1992, Vol. 4, No. 3, 211–242.

[14] ISRAELSEN D., *Collectives, communes, and incentives*, J. Compar. Econ., 1980, Vol. 4, 99–124.

[15] JAMES L.D., LEE R.R., *Economics of Water Resource Planning*, McGraw-Hill, 1971.

[16] KALETA M., *Selected models and optimization algorithms for local power energy market*, Ph.D. Thesis, Institute of Control and Computation Engineering, Warsaw University of Technology, 2004.

[17] KALETA M., *Subsidize-free cost allocation method for infrastructure market game*, Proceedings of 13th IEEE International Conference on Methods and Models in Automation and Robotics, 2007, 1171–1175.

[18] KALETA M., TOCZYŁOWSKI E., *A method for system constraint cost evaluation on the balancing energy market*, Proceedings of APE 2005 XII International Scientific Conference on Present-day Problems of Power Engineering, III, 2005, 163–170.

[19] KALETA M. TOCZYŁOWSKI E., *Parametric analysis of resource cost allocation in market balancing* (in Polish), Z. Bubnicki, R. Kulikowski, J. Kacprzyk: XV Krajowa Konferencja Automatyki, III, 2005, 341–346.

[20] KRUŚ L., BRONISZ P., *Cooperative game solution concepts to a cost allocation problem*, European Journal of Operational Research, 2000, Vol. 122, No. 2, 258–271.

[21] LENG M.M., PARLAR M., *Game theoretic applications in supply chain management*: A review, Infor., 2005, Vol. 43, No. 3, 187–220.

[22] LIMA J.W.M., *Allocation of transmission fixed charges:* an overview, IEEE Trans. Power Syst., 1996, Vol. 11, No. 4, 1409–1418.

[23] LIMA J., PEREIRA M., PEREIRA J., *An integrated framework for cost allocation in a multi-owned transmission system*, IEEE Trans. Power Syst., 1995, Vol. 10, No. 2, 971–977.

[24] LOEHMAN E., WHINSTON A., *An axiomatic approach for cost allocation of public investement*, Public Finanse Quartely, 1974, Vol. 2, No. 2, 236–251.

[25] MAS-COLELL A., *Remarks on the game theoretic analysis of a simple distribution of surplus problem*, Int. J. Game Theory, 1980, Vol. 9, No. 3, 125–140.

[26] MILLS H.D., *Marginal values of matrix games and linear programming*, Linear Inequalities and Related Systems, 1956, 183–193.

[27] OWEN G., *On the core of linear production games*, Mathematical Programming, 1975, Vol. 9, No.1, 358–370.

[28] RAMSEY F., *A contribution to the theory of taxation*, Economic Journal, 1927, Vol. 37, No. 45, 47–61.

[29] ROEMER J., *A public ownership resolution of the tragedy of the commons*, Tech. Rep. University of California, 1988.

[30] SAMET D., TAUMAN Y., ZANG I., *An application of the Aumann–Shapley prices for cost allocation in transportation problems*, Mathematics of Operations Research, 1984, Vol. 9, No. 1, 25–42.

[31] SCARF H., *Notes on the core of a productive economy*, Contributions to Mathematical Economics in Honor of Gerard Debreu, North-Holland, 1986.

[32] SCHMEIDLER D., *The nucleous of a characteristic function game*, SIAM Journal on Applied Mathematics, 1969, Vol. 17, 1163–1170.

[33] SEN A., *Labour allocation in a cooperative enterprise*, Rev. Econ. Stud., 1966, Vol. 33, 361–371.

[34] SHANG W., VOLIJ O., *Transmission investment cost allocation within the cooperative game framework*, Proceedings of Power Systems Conference and Exposition (PSCE'06), 2006, 1971–1977.

[35] SHAPLEY L.S., *A value for n-person games*, Contributions to the Theory of Games in Annals of Mathematics Studies, II, 1953, No. 28, 307–317.

[36] SHAPLEY L.S., *Cores of convex games*, International Journal of Game Theory, 1971, Vol. 1, No. 1, 11–26.

[37] SHAPLEY L.S., SHUBIK M., *The assignment game I: The Core*, International Journal of Game Theory, 1971, Vol. 1, No. 1, 111–130.

[38] SHENKER S., *Efficient network allocations with selfish users*, Performance, 1990.

[39] SHENKER S., *Making greed work in networks: A game-theoretic analysis of gateway service disciplines*, IEEE ACM Trans. Networking, 1995, Vol. 3, No. 45, 819–831.

[40] SKORIN-KAPOV D., SKORIN-KAPOV J., *Threshold based discounting networks: The cost allocation provided by the nucleolus*, European Journal of Operational Research, 2005, Vol. 166, No. 1, 154–159.

[41] STAMTSIS G.C., ERLICH I. *Use of cooperative game theory in power system fixed-cost allocation*, IEE Proc.-Gener. Transm. Distrib., 2004, Vol. 151, No. 3.

[42] TAN X.H., LIE T.T., *Allocation of transmission loss cost using cooperative game theory in the context of open transmission access*, IEEE, Power Engineering Society Winter Meeting, 2001, 3, 1215–1219.

[43] TAUMAN Y. *The Aumann-Shapley prices: a survey*, The Shapley Value: Essays in Honor of Lloyd S. Shapley., Cambridge University Press.

[44] TOCZYŁOWSKI E., *Optimisation of turnover of many commodities linked by resources* (in Polish), Zesz. Nauk. Polit. Śląskiej, seria Automatyka, 2000, Vol. 130, 171–179.

[45] WEBER R., *Probabilistic values for games*, The Shapley Value, Cambridge University Press, 1988.

[46] YOUNG H.P., *Cost Allocation: Methods, Principles, Applications*, Elseviers Science Publishers B.V., 1985.

Metaalgorytm alokacji kosztów dla gier LP i GLP

Rozważono zagadnienie alokacji kosztów w ramach klasy problemów modelowanych jako GLPG (***G**eneralized **L**inear **P**rogramming **G**ames*) – uogólnione gry kooperatywne bazujące na zadaniach programowania liniowego. Zakładamy, że grupa agentów uczestniczy we wspólnym przedsięwzięciu, a każdy agent określa pewne wymagania związane z jego uczestnictwem. Wówczas pojawia się problem podziału łącznych kosztów realizacji przedsięwzięcia na każdego z uczestniczących agentów przy zapewnieniu zdefiniowanych ograniczeń tych agentów.

W artykule sformułowano ogólny szkielet algorytmu do wyznaczania alokacji łącznych kosztów, oparty na klasie metod alokacji generowanych przez tzw. ścieżki. Proponowane podejście jest uogólnieniem pewnych ważnych mechanizmów alokacji, bazujących na teorii gier kooperatywnych, w tym na wycenie Aummana–Shapleya. Zgodnie z naszą wiedzą dotychczas nie został opracowany żaden wydajny algorytm umożliwiający wyznaczanie alokacji dla problemów modelowanych jako gry kooperatywne bazujące na programowaniu liniowym (LPG i GLPG). Spotykane w literaturze podejścia polegają na obliczeniach niedokładnych przy jednocześnie wymaganym większym nakładzie obliczeniowym.

Słowa kluczowe: *alokacja kosztów, gry kooperatywne, gra bazująca na programowaniu liniowym*