Kimms, Alf; Drechsel, Julia

Article

Cost Sharing under Uncertainty: An Algorithmic Approach to Cooperative Interval-Valued Games

BuR - Business Research

Provided in Cooperation with:
VHB - Verband der Hochschullehrer für Betriebswirtschaft, German Academic Association of Business Research

Suggested Citation: Kimms, Alf; Drechsel, Julia (2009) : Cost Sharing under Uncertainty: An Algorithmic Approach to Cooperative Interval-Valued Games, BuR - Business Research, ISSN 1866-8658, VHB - Verband der Hochschullehrer für Betriebswirtschaft, German Academic Association of Business Research, Göttingen, Vol. 2, Iss. 2, pp. 206-213, http://dx.doi.org/10.1007/BF03342711

BuR – Business Research
Official Open Access Journal of VHB
Verband der Hochschullehrer für Betriebswirtschaft e.V.
Volume 2 | Issue 2 | December 2009 | 206–213

Cost Sharing under Uncertainty: An Algorithmic Approach to Cooperative Interval-Valued Games

Alf Kimms, *Mercator School of Management, University of Duisburg-Essen, Germany, E-mail: alf.kimms@uni-due.de*
Julia Drechsel, *Mercator School of Management, University of Duisburg-Essen, Germany, E-mail: julia.drechsel@uni-due.de*

Abstract
Recently, Branzei, Dimitrov, and Tijs (2003) introduced cooperative interval-valued games. Among other insights, the notion of an interval core has been coined and proposed as a solution concept for interval-valued games. In this paper we will present a general mathematical programming algorithm which can be applied to find an element in the interval core. As an example, we discuss lot sizing with uncertain demand to provide an application for interval-valued games and to demonstrate how interval core elements can be computed. Also, we reveal that pitfalls exist if interval core elements are computed in a straightforward manner by considering the interval borders separately.

Keywords: cooperative game theory, core, interval-valued games, mathematical programming, lot-sizing

Manuscript received June 22, 2009, accepted by Karl Inderfurth (Operations & Information Systems) November 17, 2009.

1 A Primer on Cooperative Interval-Valued Games

In the classical literature a cooperative game is defined by a pair (N, c) where $N = \{1, \ldots, n\}$ is an index set of players and $c : 2^N \rightarrow R$ is a characteristic function which assigns to every coalition $S \in 2^N$ a value $c(S)$ (with $c(\emptyset) = 0$). Let us assume that c is a cost function ("the lower the better"), but in settings where c is a benefit function the following material can be straightforwardly adapted. A solution of the cooperative game (with transferable utilities) is a distribution of the cost shares, i.e. a cost allocation $\pi_i \in R$ for every player $i \in N$. One of the most prominent solution concepts was introduced by Gillies (1959) and is known as the core. The concept of the core defines cost allocations which are efficient and coalitionally rational (stable). Formally, the set of core allocations can be specified as follows:

$$(1)\quad C(N,c) = \{\pi \in R^n | \sum_{i \in N} \pi_i = c(N) \text{ and } \sum_{i \in S} \pi_i \leq c(S) \text{ for all } S \subset N, S \neq \emptyset\}.$$

Branzei, Dimitrov, and Tijs (2003) have studied bankruptcy situations and introduced cooperative interval-valued games in this context. Interval-valued games can be seen as a means to handle uncertain outcomes. A theory of interval-valued games was developed by Alparslan-Gök, Branzei, and Tijs (2008a, 2008b) as well as by Alparslan-Gök, Miquel, and Tijs (2009). See also Branzei, Tijs, and Alparslan-Gök (2008) for a survey. A cooperative interval-valued game is defined by a pair (N, c^I) where N is an index set of players as before, and $c^I : 2^N \rightarrow I(R)$ is a characteristic (cost) function which assigns to every coalition $S \in 2^N$ a closed interval $c^I(S) = [\underline{c}^I(S); \bar{c}^I(S)]$ (with $c^I(\emptyset) = [0; 0]$) where $I(R)$ is the set of all closed intervals in R. It is easy to see that classical cooperative games are a special case of cooperative interval-valued games where $\underline{c}^I(S) = \bar{c}^I(S)$ for all coalitions $S \subseteq N$. It should also be remarked that vector-valued games (see Fernandez, Hinojosa, and Puerto 2002) are somewhat different to interval-valued games.
To define a core variant that applies to interval-

BuR – Business Research
Official Open Access Journal of VHB
Verband der Hochschullehrer für Betriebswirtschaft e.V.
Volume 2 | Issue 2 | December 2009 | 206–213

valued games, the following notation turns out to be helpful. The sum of two intervals $I = [\underline{I}; \bar{I}]$ and $J = [\underline{J}; \bar{J}]$ is defined to be $I + J = [\underline{I} + \underline{J}; \bar{I} + \bar{J}]$. The interval I is called weakly better than interval J ($I \preceq J$ or $J \succeq I$), iff $\underline{I} \leq \underline{J}$ and $\bar{I} \leq \bar{J}$.

Alparslan-Gök, Branzei, and Tijs (2008a) define the interval core of a cooperative interval-valued game as follows:

$$(2) \quad C^I(N, c^I) = \{(I_1, \ldots, I_n) \in I(R)^n \mid \sum_{i \in N} I_i = c^I(N) \text{ and } \sum_{i \in S} I_i \preceq c^I(S) \text{ for all } S \subset N, S \neq \emptyset\}.$$

These authors, however, do not provide an algorithm to compute a core element of interval-valued games.

Now we have introduced the theoretical background that is necessary for understanding our approach. Our contribution is an algorithm which can be used to compute an interval core element. Besides this, we discuss pitfalls when implementing a more simple procedure based on an already known algorithm for core element computation in cases of independent determination of interval borders. The rest of the paper is structured as follows. In Section 2 we specify the straightforward procedure with separate interval border determination that aims to find an element in the interval core (or detects that the interval core is empty). Section 3 introduces a lot-sizing problem with uncertain demand as an example for interval-valued games. Section 4 reveals that some odd situations may occur if the straightforward approach is applied and left and right interval borders are used independently to compute interval core elements. Thus, a modification of the proposed algorithm will be presented to avoid such problems. A computational study in Section 5 demonstrates that this modified algorithm is necessary in many cases and can indeed be successfully applied. Concluding remarks in Section 6 finish the paper.

2 Computing Interval Core Elements

To compute an element in the interval core, we will make use of a procedure by Drechsel and Kimms (2007) which computes a core element. This procedure is employed as a subroutine. To see how this works we start with a reformulation of interval-valued games and the interval core. An interval-valued game can be specified by a triple $(N, \underline{c}^I, \bar{c}^I)$ where the notation has the same meaning as above. Two sets of extreme cost allocations can then be defined as

$$(3) \quad \underline{C}^I(N, \underline{c}^I) = \{\pi \in R^n \mid \sum_{i \in N} \pi_i = \underline{c}^I(N) \text{ and } \sum_{i \in S} \pi_i \leq \underline{c}^I(S) \text{ for all } S \subset N, S \neq \emptyset\}$$

and

$$(4) \quad \bar{C}^I(N, \bar{c}^I) = \{\pi \in R^n \mid \sum_{i \in N} \pi_i = \bar{c}^I(N) \text{ and } \sum_{i \in S} \pi_i \leq \bar{c}^I(S) \text{ for all } S \subset N, S \neq \emptyset\}.$$

If we choose $(\underline{I}_1, \ldots, \underline{I}_n) \in \underline{C}^I(N, c)$ and $(\bar{I}_1, \ldots, \bar{I}_n) \in \bar{C}^I(N, c)$ then it follows immediately from the definitions of c^I, I_i and $C^I(N, c^I)$ that $([\underline{I}_1; \bar{I}_1], \ldots, [\underline{I}_n; \bar{I}_n]) \in C^I(N, c^I)$ — the reverse is true as well. $\underline{C}^I(N, \underline{c}^I)$ and $\bar{C}^I(N, \bar{c}^I)$ define the cores of the cooperative games $(N, \underline{c}^I)$ and $(N, \bar{c}^I)$, respectively. This is along the lines of Remark 3.1 in the paper by Alparslan-Gök, Branzei, and Tijs (2008a), who state that the time complexity of algorithms for the computation of elements of the interval core is the same as the time complexity of related algorithms for computing elements of the core of a traditional game. This suggests that an element in the interval core can be easily computed if one can compute an element in the core.

The procedure of Drechsel and Kimms (2007) can be applied to find core elements for the games $(N, \underline{c}^I)$ and $(N, \bar{c}^I)$ separately. To be self-contained, this procedure will be reviewed briefly. To ease the notation, assume that a core element of a game (N, c) shall be computed. The definition of the core specifies a constraint-satisfaction problem where the number of constraints is in the order of 2^n. Hence, Drechsel and Kimms (2007) suggested to run a row-generation procedure.

The master problem is a linear program of the following form where $\mathcal{S}$ is a (small) subset of coalitions for which the core defining inequality is explicitly taken into account.

BuR – Business Research
Official Open Access Journal of VHB
Verband der Hochschullehrer für Betriebswirtschaft e.V.
Volume 2 | Issue 2 | December 2009 | 206–213

Master problem $MP(\mathcal{S})$:

$$\min v \tag{5}$$

s.t.

$$\sum_{i \in N} \pi_i = c(N) \tag{6}$$

$$\sum_{i \in S} \pi_i - v \leq c(S) \qquad S \in \mathcal{S} \tag{7}$$

$$\pi_i \in R \qquad i \in N \tag{8}$$

$$v \geq 0 \tag{9}$$

The iterative row-generation procedure can be outlined as follows:

1. Define a small initial set $\mathcal{S}$, e.g. $\mathcal{S} = \{\{1\}, \ldots, \{n\}\}$.
2. Solve the linear program $MP(\mathcal{S})$ optimally.
3. If $v > 0$ then stop. The game instance has an empty core.
4. Otherwise, find a coalition $S' \notin \mathcal{S}$ $(S' \neq \emptyset)$ such that $\sum_{i \in S'} \pi_i > c(S')$. Since S' may not be unique, we suggest looking for a coalition S' that violates the core defining inequality most (i.e. $\sum_{i \in S'} \pi_i - c(S')$ is maximized) — subproblem $\hat{SP}(\pi)$.
5. If no such coalition S' can be found ($\hat{SP}(\pi)$ has a non-positive optimum objective function value) then stop. The current values π_i define a core allocation.
6. Otherwise, update $\mathcal{S} = \mathcal{S} \cup \{S'\}$. Return to Step 2.

The subproblem to be solved in Step 4 of the algorithm is the only problem-specific part in this approach. In a subsequent section we will provide the details of it by means of a specific application. It should be noted that this procedure finds a rather arbitrary element in the core (if the core is not empty). Drechsel and Kimms (2007) have discussed several modifications to the procedure which yield core elements that can be considered as more fair than an arbitrary core element. For the sake of simplicity, we will not repeat these modifications here, but we would like to emphasize that these modifications can be straightforwardly integrated into the context of interval-valued games if desired.

3 An Application: Lot-Sizing with Uncertain Demand

3.1 The Wagner-Whitin Problem

The starting point of our example is a well-known lot-sizing problem which was introduced by Wagner and Whitin (1958) and which can be described as follows: A single decision maker has to make order decisions for a single item. Given a planning horizon of T time periods, a demand d_t in period t has to be met without backlogging and without shortages. In order to meet the demand in period t one may place an order in t or before. If the order is placed before, the ordered items must be stored in inventory. If a fixed cost s_t is incurred whenever an order is placed in period t and a unit holding cost h_t is charged for every item in stock at the end of period t, then we face the classical trade-off between saving fixed costs versus saving holding costs — a lot-sizing problem occurs. In addition to that a unit ordering cost p_t may be incurred for each item being ordered. The decision to be made is q_t, the quantity to be ordered in period t. Depending on the order quantities and the demand we have Inv_t units of the item in stock at the end of a period t (Inv_0 is the initial stock — a given value). To formulate a mixed-integer program a further decision variable x_t may be used to indicate if an order is placed in period t or not, and a parameter M which represents a large number may be used for technical reasons, as we see below:

$$\min \sum_{t=1}^{T} (s_t x_t + h_t Inv_t + p_t q_t) \tag{10}$$

s.t.

$$Inv_t = Inv_{t-1} + q_t - d_t \qquad t = 1, \ldots, T \tag{11}$$

$$q_t \leq M x_t \qquad t = 1, \ldots, T \tag{12}$$

$$q_t, Inv_t \geq 0 \qquad t = 1, \ldots, T \tag{13}$$

$$x_t \in \{0, 1\} \qquad t = 1, \ldots, T \tag{14}$$

The objective (10) is to minimize the sum of fixed and quantity-dependent ordering costs and holding costs. The inventory balance (11) states that at the end of a period we have in stock what was there at the beginning of the period plus what was ordered in that period minus period demand. If the order quantity is positive in period t then the indicator variable x_t must be set to one as stated by (12). The domain of the decision variables is

BuR – Business Research
Official Open Access Journal of VHB
Verband der Hochschullehrer für Betriebswirtschaft e.V.
Volume 2 | Issue 2 | December 2009 | 206–213

specified in (13) and (14). Note that due to $Inv_t \geq 0$ all demand must be fulfilled right on time.

3.2 A Cooperative Lot-Sizing Problem

Cooperative versions of the Wagner-Whitin-related problem where several players (the set N) may act together have been studied by Chen and Zhang (2007), Drechsel and Kimms (2007), Guardiola, Meca, and Puerto (2006), and Van den Heuvel, Borm, and Hamers (2007). A model formulation closely related to the single-player model shown above can be given, simply by replacing the demand value d_t with the total demand of all players $d_t(N) = \sum_{i \in N} d_{it}$ where d_{it} denotes the demand of player i in period t. The characteristic function value $c(S)$ for $S \subseteq N$ is defined to be the optimum objective function value (10) using $d_t(S)$ as the demand in (11).

As shown by Drechsel and Kimms (2007) one can apply the row-generation procedure for finding a core element in the game (N, c) to handle instances with up to 150 players. The subproblem that has to be solved in Step 4 of the procedure is the following optimization problem.

Subproblem $\hat{SP}(\pi)$:

$$(15) \quad \max \left(-o + \sum_{i \in N} \pi_i z_i \right)$$

s.t.

$$(16) \quad Inv_t = Inv_{t-1} + q_t - \sum_{i \in N} d_{it} z_i \qquad t = 1, \ldots, T$$

$$(17) \quad q_t \leq M x_t \qquad t = 1, \ldots, T$$

$$(18) \quad o = \sum_{t=1}^{T} (s_t x_t + h_t Inv_t + p_t q_t)$$

$$(19) \quad q_t, Inv_t \geq 0 \qquad t = 1, \ldots, T$$

$$(20) \quad x_t \in \{0, 1\} \qquad t = 1, \ldots, T$$

$$(21) \quad z_i \in \{0, 1\} \qquad i \in N$$

$$(22) \quad o \geq 0$$

Note that a coalition S' to be considered in the master problem is found if the optimum objective function value of the subproblem is positive. S' is defined by the values of the z_i variables ($z_i = 1$ indicates $i \in S'$) and $c(S') = o$. Note also that Drechsel and Kimms (2007) have tested other subproblem formulations, and that selecting a coalition S' by means of subproblem $\hat{SP}(\pi)$ turned out to require fewer iterations of the row-generation procedure and less total computation time than other selection rules.

It should be emphasized that the proposed row-generation approach to compute a core element is very general and can be applied whenever the characteristic function is implicitly given, i.e. characteristic function values result from an optimization problem. Drechsel and Kimms (2008), for instance, have shown that for much more complex lot-sizing situations with capacity constraints and transshipments among the players basically the very same approach is applicable. Only the subproblem needs to be reformulated, because this is the problem-specific part of the algorithm. How to formulate such subproblems for different applications cannot be specified in general, but it requires no more skills than formulating the underlying optimization problem as a mathematical program.

As a side remark, it should be emphasized that many operations management planning problems have to be solved repeatedly with a rolling horizon. Lot sizing is one example. Our example is therefore stylized by ignoring (uncertain demand) data beyond the period T and we admit that our focus is confined to interval-valued data. We also do not consider the possibility to condition future lot-sizing decisions on realized past demands like in stochastic dynamic optimization procedures. To the best of our knowledge, such simplifying assumptions are also made by all authors who have considered cost sharing based on multi-period optimization problems. Hence, one should be aware of the fact that dynamic settings where uncertainty is not revealed at once and the consequences on cost sharing have not been studied yet.

3.3 Cooperative Lot-Sizing with Uncertain Demand

A source for uncertainty in a lot-sizing problem is the demand of a player. Let us assume that for each player i the demand in a certain period is specified by an interval $[\underline{d}_{it}; \bar{d}_{it}]$ instead of a single number d_{it}. Consequently, the total demand $d_t(N)$ in a period t falls into the interval $[\underline{d}_t(N); \bar{d}_t(N)] = [\sum_{i \in N} \underline{d}_{it}; \sum_{i \in N} \bar{d}_{it}]$. From these interval borders we can derive characteristic functions $\underline{c}^I$ and $\bar{c}^I$ by inserting $\underline{d}_t(S)$ and $\bar{d}_t(S)$, respectively, into (11) for $S \subseteq N$. With this in mind, we can compute an element in the interval core as described in a previous section.

BuR – Business Research
Official Open Access Journal of VHB
Verband der Hochschullehrer für Betriebswirtschaft e.V.
Volume 2 | Issue 2 | December 2009 | 206–213

4 Phenomena of Interval Cores

4.1 "Pay Less for More"

As some researchers have pointed out already (see, e.g., Young 1985), core allocations are not monotonic. This effect causes some problems in the context of interval cores as we will show by means of an example. Consider an example with $n = 3$ and $T = 6$. Let the demand be $\underline{d}_{it} = 10$ and $\bar{d}_{it} = 15$, respectively, for all i and t. Suppose the following cost data: holding cost coefficients $h_t = 5$ for all t, setup cost coefficients $s_t = 100$ for all t, and production cost coefficients $p_t = 1$ for all t. The characteristic functions $\underline{c}^I$ and $\bar{c}^I$ are provided in Table 1.

Table 1: The Characteristic Functions of the Interval-Valued Game

S	$\emptyset$	$\{1\}$	$\{2\}$	$\{3\}$
$\underline{c}^I(S)$	0	510	510	510
$\bar{c}^I(S)$	0	615	615	615
S	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
$\underline{c}^I(S)$	720	720	720	780
$\bar{c}^I(S)$	780	780	780	870

Apparently, we have $\bar{c}^I(S) \geq \underline{c}^I(S)$ in this example. This, by the way, is always true for lot-sizing problems (with non-negative cost data), because additional demand cannot decrease the (total) cost to fulfill the demand. Since the characteristic functions $\bar{c}^I$ and $\underline{c}^I$ define core cost allocations $\pi_i(\bar{c}^I)$ and $\pi_i(\underline{c}^I)$, respectively, we would expect that $\pi_i(\bar{c}^I) \geq \pi_i(\underline{c}^I)$ is true for all players i. In other words, we would expect that increasing the demand of a player does not lead to a lower cost assignment, i.e. the principle of monotonicity holds. But this is not true in general. For example, $\pi(\underline{c}^I) = (210, 510, 60)$ is a core cost allocation for the game $(N, \underline{c}^I)$. On the other hand, $\pi(\bar{c}^I) = (165, 615, 90)$ is a core cost allocation for the game $(N, \bar{c}^I)$. As we can see, a lower cost share would be assigned to player 1 (165 instead of 210) although his demand increased from 10 to 15 in every period.

Young (1985) has discussed the issue of monotonicity and showed that the Shapley value is the unique symmetric allocation procedure that is (strongly) monotonic. But using the Shapley value as an interval border is not consistent with the idea of the interval core, because the Shapley value need not be in the core if the game is not concave (see Shapley 1971).

As a consequence of the phenomenon we may observe intervals $I_i = [\underline{I}_i; \bar{I}_i]$ with $\underline{I}_i > \bar{I}_i$. This, by the way, is not valid for lot-sizing problems only, but to all kinds of applications, because the non-monotonicity of core allocations is problem independent.

4.2 Pitfalls

It seems to be plausible to interpret an element from the interval core as a promise to the players such that if $(I_1, \ldots, I_n) \in C^I(N, c^I)$ is chosen in advance, an ex post (i.e. after the uncertainty has been resolved) core cost allocation with $\pi_i \in I_i$ can be found, because the two extreme scenarios have been considered to define the interval core. Hence, the problem of finding a core element ex post seems to be of the following form:

$$(23) \quad C|_{(I_1,\ldots,I_n)}(N,c) = \{\pi \in R^n | \sum_{i \in N} \pi_i = c(N) \text{ and } \sum_{i \in S} \pi_i \leq c(S) \text{ for all } S \subset N, S \neq \emptyset \text{ and } \pi_i \in I_i\}.$$

Note that this interpretation is meaningful iff the underlying (ex post) characteristic function c is monotone over the interval defined by c^I and coincides with $\underline{c}^I$ and $\bar{c}^I$ at the interval borders. This interpretation, however, is wrong as we will show be means of the example from Subsection 4.1.

Recall that in the example the following intervals were computed: $I_1 = [210; 165]$, $I_2 = [510; 615]$, and $I_3 = [60, 90]$. Assume that ex post the following demand data can be observed: $d_{1t} = 15$ for all t, $d_{2t} = 10$ for all t, and $d_{3t} = 15$ for all t. As we can see from Table 1, this leads to $c(\{1\}) = 615$, $c(\{2\}) = 510$, and $c(\{3\}) = 615$. The cost of the grand coalition is $c(\{1,2,3\}) = 840$. The definition of $C|_{(I_1,\ldots,I_n)}(N,c)$ requires $\pi_2 = 510$. Hence, $\pi_1 + \pi_3 = 840 - 510 = 330$. But this is not possible, because of $\pi_1 \leq 210$ and $\pi_3 \leq 90$. Hence, the described approach leads to unreasonable results.

4.3 A Proper Definition of Interval-Valued Cores

To avoid such problems with interval cores, we need proper intervals, i.e. we need to compute interval borders $\pi_i(\underline{c}^I)$ and $\pi_i(\bar{c}^I)$ such that $\pi_i(\underline{c}^I) \leq$

BuR – Business Research
Official Open Access Journal of VHB
Verband der Hochschullehrer für Betriebswirtschaft e.V.
Volume 2 | Issue 2 | December 2009 | 206–213

$\pi_i(\bar{c}^I)$ comes out. Such a result can be reached be computing the values $\pi_i(\underline{c}^I)$ and $\pi_i(\bar{c}^I)$ simultaneously by solving a modified master problem $MP^I(\underline{\mathcal{S}}, \bar{\mathcal{S}})$:

$$\min \underline{v} + \bar{v} \tag{24}$$

s.t.

$$\sum_{i \in N} \pi_i(\underline{c}^I) = \underline{c}^I(N) \tag{25}$$

$$\sum_{i \in S} \pi_i(\underline{c}^I) - \underline{v} \le \underline{c}^I(S) \qquad S \in \underline{\mathcal{S}} \tag{26}$$

$$\sum_{i \in N} \pi_i(\bar{c}^I) = \bar{c}^I(N) \tag{27}$$

$$\sum_{i \in S} \pi_i(\bar{c}^I) - \bar{v} \le \bar{c}^I(S) \qquad S \in \bar{\mathcal{S}} \tag{28}$$

$$\pi_i(\underline{c}^I) \le \pi_i(\bar{c}^I) \qquad i \in N \tag{29}$$

$$\pi_i(\underline{c}^I), \pi_i(\bar{c}^I) \in \boldsymbol{R} \qquad i \in N \tag{30}$$

$$\underline{v}, \bar{v} \ge 0 \tag{31}$$

This master problem can be solved by the iterative row-generation algorithm as follows:

1. Define small initial sets $\underline{\mathcal{S}}$ and $\bar{\mathcal{S}}$, e.g. $\underline{\mathcal{S}} = \bar{\mathcal{S}} = \{\{1\}, \ldots, \{n\}\}$.
2. Solve the linear program $MP^I(\underline{\mathcal{S}}, \bar{\mathcal{S}})$ optimally.
3. If $\underline{v} + \bar{v} > 0$ then stop. The game instance has an empty interval core.
4. Otherwise:
   (a) Find a coalition $\underline{S}' \notin \underline{\mathcal{S}}$ $(\underline{S}' \neq \emptyset)$ such that $\sum_{i \in \underline{S}'} \pi_i(\underline{c}^I) > \underline{c}^I(\underline{S}')$. Since $\underline{S}'$ may not be unique, we suggest looking for a coalition $\underline{S}'$ that violates the core defining inequality most (i.e. $\sum_{i \in \underline{S}'} \pi_i(\underline{c}^I) - \underline{c}^I(\underline{S}')$ is maximized) — subproblem $\underline{\hat{SP}}(\pi(\underline{c}^I))$. Update $\underline{\mathcal{S}} = \underline{\mathcal{S}} \cup \{\underline{S}'\}$, if such a coalition $\underline{S}'$ was found.
   (b) Find a coalition $\bar{S}' \notin \bar{\mathcal{S}}$ $(\bar{S}' \neq \emptyset)$ such that $\sum_{i \in \bar{S}'} \pi_i(\bar{c}^I) > \bar{c}^I(\bar{S}')$. Since $\bar{S}'$ may not be unique, we suggest looking for a coalition $\bar{S}'$ that violates the core defining inequality most (i.e. $\sum_{i \in \bar{S}'} \pi_i(\bar{c}^I) - \bar{c}^I(\bar{S}')$ is maximized) — subproblem $\hat{\bar{SP}}(\pi(\bar{c}^I))$. Update $\bar{\mathcal{S}} = \bar{\mathcal{S}} \cup \{\bar{S}'\}$, if such a coalition $\bar{S}'$ was found.
5. If neither $\underline{S}'$ nor $\bar{S}'$ can be found then stop. The current values $\pi_i(\underline{c}^I)$ and $\pi_i(\bar{c}^I)$ define an interval core element.
6. Return to Step 2.

Recall the numerical example from Section 4.1. The presented algorithm yields the following allocation in the interval-valued core: $\pi(\underline{c}^I) = (210, 60, 510)$ and $\pi(\bar{c}^I) = (210, 90, 570)$.

5 Computational Study

To test the proposed algorithm from Section 4.3 we have implemented it with AMPL/CPLEX version 10.0.0 on an Intel Celeron PC with 2 GHz. We used random instances taken from Test Bed 1 in Drechsel and Kimms (2007) to define the game $(N, \underline{c}^I)$: random integers were drawn with uniform distribution from certain intervals ($d_{it} \in [0, 20]$, $s_t \in [0, 200]$, $h_t \in [0, 10]$, and $p_t \in [0, 15]$), $T = 6$, and $n \in \{5, 10, 15, 20, 25, 30\}$. For each number of players, a total of 15 random instances was constructed which gives a total of $6 \times 15 = 90$ instances. These instances are provided as an electronic companion to this paper as part of the online version.[1] The files are in ASCII format and can be used by the commercial software CPLEX. The game $(N, \bar{c}^I)$ was constructed by copying the parameter values from the corresponding game $(N, \underline{c}^I)$. Only demand values were chosen differently in the following way. Suppose $\underline{d}_{it}$ is the chosen demand in the corresponding game $(N, \underline{c}^I)$. The demand d_{it} for the game $(N, \bar{c}^I)$ is a random integer drawn with uniform distribution from $[\underline{d}_{it}, 20]$.

The average number of iterations (as well as the minimum/maximum number of iterations over 15 random instances) for the algorithm can be seen in Table 2. The table presents the results regarding number of iterations for the two subproblems $\underline{\hat{SP}}$ and $\hat{\bar{SP}}$ that are part of Step 4 of the algorithm. The numbers differ for most of the instances — of course, the bigger number also denotes the total number of needed iterations. Note that in the worst case the number of iterations could have been in the order of 2^n which practically never happens from our experience. The maximum computational time (measured in CPU-seconds) to determine an element in the interval-valued core was less than a minute for all instances $n \le 20$ and not more than five minutes for the larger instances.

To compare the computational burden with the straightforward approach from Section 2 — the

[1] See www.business-research.org.

BuR – Business Research
Official Open Access Journal of VHB
Verband der Hochschullehrer für Betriebswirtschaft e.V.
Volume 2 | Issue 2 | December 2009 | 206–213

Table 2: The Number of Iterations Required to Find an Interval Core Element

n	5	10	15
$\underline{\hat{SP}}$	4.87 (4/7)	20.47 (14/40)	146.27 (25/293)
$\bar{\hat{SP}}$	4.73 (4/6)	21.27 (12/60)	146.27 (16/293)
n	**20**	**25**	**30**
$\underline{\hat{SP}}$	406.73 (110/1036)	1044.60 (165/2364)	1043.53 (30/3946)
$\bar{\hat{SP}}$	403.80 (110/1037)	1039.80 (165/2360)	1054.53 (30/3941)

one which may lead to the problems discussed in Section 4.2 — we also provide the average number of iterations of the straightforward approach in Table 3. As one can see the straightforward approach terminates much faster on average (the run-time effort was less than one minute for each instance). Hence it might be worthwhile to run the straightforward approach first and test if the computed interval borders indeed show the odd behaviour discussed in Section 4.2. If this is not the case, then we have found an interval core element. Otherwise, we need to run the algorithm from Section 4.3 which is necessary in most cases as we can see in Table 4. Table 4 shows the number of instances in the test bed (a total of 15 instances per value of n) where the algorithm from Section 2 yields an improper interval for at least one player.

Table 3: The Average Number of Iterations Required by the Straightforward Approach from Section 2

n	5	10	15
$(N, \underline{c}^I)$	4.0	10.5	15.3
$(N, \bar{c}^I)$	4.0	9.3	14.0
n	**20**	**25**	**30**
$(N, \underline{c}^I)$	61.8	29.5	50.0
$(N, \bar{c}^I)$	19.0	24.4	29.0

Table 4: The Number of Instances where the Algorithm from Section 2 Yields Improper Intervals

n	5	10	15	20	25	30
	14	13	12	4	11	11

6 Conclusions

In recent years a new class of cooperative games, namely interval-valued games, has been invented to handle situations with an uncertain outcome. The range of the outcome is specified by a closed interval. A theory for interval-valued games has been developed and solution concepts from classical cooperative game theory have been adapted to this new setting. The core which is one of the most prominent solution concepts in classical cooperative game theory evolved to become the interval core.

Remark 3.1 in a paper by Alparslan-Gök, Branzei, and Tijs (2008a) states that the time complexity of algorithms for computing elements of interval cores equals the time complexity of corresponding algorithms for computing core elements in a traditional game. This statement suggests to compute an element in the interval core by making use of a procedure to find a core element. However, such an algorithm has not been provided by these authors. Our contribution is to provide a general mathematical programming approach which can be used to compute an element in the interval core by making use of a procedure to find core elements proposed by Drechsel and Kimms (2007) as a subroutine. The approach is general, because the characteristic function values which define the lower and upper bounds of the outcome interval may result from solving general optimization problems. We have demonstrated that interval-valued games are a consequence of uncertain parameter values of the optimization problem where the uncertainty of these parameter values is specified by intervals. With the aid of a lot-sizing example we have demonstrated the applicability of our approach by means of a small computational study.

In addition to providing an algorithm we have also contributed to the discussion of interval cores by pointing to a pitfall. A phenomenon known as non-monotonicity exists, i.e. players may be better off in a cooperation, if their standalone situation is getting worse (in terms of objective function values). In the lot-sizing example, for instance, we have demonstrated that a player with more demand than before (increasing costs) may receive a lower cost share than before. A straightforward interpretation of an interval core element in the sense of cost limits that will be assigned to certain players is not possible if the interval core element is

BuR – Business Research
Official Open Access Journal of VHB
Verband der Hochschullehrer für Betriebswirtschaft e.V.
Volume 2 | Issue 2 | December 2009 | 206–213

not selected with care. To get out of this problem, we proposed a modified algorithm which yields an element from the interval core without such deficits.

7 Acknowledgement

We thank Rodica Branzei for her comments on an early version of this paper.

References

Alparslan-Gök, Sirma Z., Rodica Branzei, and Stef H. Tijs (2008a): Cores and Stable Sets for Interval-Valued Games, Working Paper, SSRN.

Alparslan-Gök, Sirma Z., Rodica Branzei, and Stef H. Tijs (2008b): Convex Interval Games, Working Paper, SSRN.

Alparslan-Gök, Sirma Z., Sylvia Miquel, and Stef H. Tijs (2009): Cooperation under Interval Uncertainty, *Mathematical Methods of Operation Research*, 69 (1): 99-109.

Branzei, Rodica, Dinko Dimitrov, and Stef H. Tijs (2003): Shapley-like Values for Interval Bankruptcy Games, *Economics Bulletin*, 3 (9): 1-8.

Branzei, Rodica, Stef H. Tijs, and Sirma Z. Alparslan-Gök (2008): Cooperative Interval Games: A Survey, Working Paper, http://www.3.iam.metu.edu.tr/iam/images/8/80/survey.pdf (Access Date: 2009-12-03).

Chen, Xin and Jiawei Zhang (2007): Duality Approches to Economic Lot-Sizing Games, Working Paper, SSRN.

Drechsel, Julia and Alf Kimms (2007): Computing Core Allocations in Cooperative Games with an Application to Procurement Problems, Working Paper, University of Duisburg-Essen.

Drechsel, Julia and Alf Kimms (2008): Solutions and Fair Cost Allocations for Cooperative Lot Sizing with Transshipments and Scarce Capacities, Working Paper, University of Duisburg-Essen.

Fernandez, Francisco R. , Miguel A. Hinojosa, and Justo Puerto (2002), Core Solutions in Vector-Valued Games, *Journal of Optimization Theory and Applications*, 112 (2): 331-360.

Gillies, Donald B. (1959): Solutions to General Non-Zero-Sum Games, in: Albert W. Tucker and R. Duncan Luce (eds.): *Contributions to the Theory of Games 4*, Princeton University Press: Princeton, 47-85.

Guardiola, Luis A. , Ana Meca, and Justo Puerto (2006): Coordination in Periodic Review Inventory Situations, Working Paper, Universidad Miguel Hernández de Elche.

Shapley, Lloyd S. (1971): Cores of Convex Games, *International Journal of Game Theory*, 1 (1): 11-26.

Van den Heuvel, Wilco, Peter Borm, and Herbert Hamers (2007): Economic Lot-Sizing Games, *European Journal of Operational Research*, 176 (2): 1117-1130.

Wagner, Harvey M. and Thomson M. Whitin (1958): Dynamic Version of the Economic Lot Size Model, *Management Science*, 5 (1): 89-96.

Young, H. Peyton (1985): Monotonic Solutions of Cooperative Games, *International Journal of Game Theory*, 14 (2): 65-72.

Biographies

Alf Kimms studied computer science and business administration at the University of Kiel. He received his Ph.D. in business administration from the University of Kiel in 1996 with a thesis on multi-level lot sizing and scheduling under the supervision of Prof. Dr. Andreas Drexl, and he earned his postdoctoral lecturer qualification from the same university in 2001 with a thesis on financial objectives for managing projects. In 2001 he became full professor at the Technical University of Freiberg and held the chair of production and logistics. In 2006 he was appointed as a full professor at the University of Duisburg-Essen to hold the chair of logistics and traffic management. Professor Kimms has received several best-paper awards.

Julia Drechsel studied industrial engineering at the Technical University of Freiberg. She received several prizes for her diploma thesis on task force deployment. From 2005 to 2009 she was a Ph.D. student under the supervision of Professor Kimms working at the University of Duisburg-Essen. Her research focus is on Supply Chain Management. Since 2009 she has been working for the company Bayer Technology Services GmbH.