

PDF hosted at the Radboud Repository of the Radboud University Nijmegen

The following full text is a publisher's version.

For additional information about this publication click this link.

<http://hdl.handle.net/2066/60207>

Please be advised that this information was generated on 2018-10-16 and may be subject to change.

Center



Discussion Paper

No. 2004–126

PROCESSING GAMES WITH SHARED INTEREST

By M. Quant, M. Meertens, J.H. Reijnierse

November 2004

ISSN 0924-7815

Processing games with shared interest

Marieke Quant^{1,2} Marc Meertens³ Hans Reijnierse²

Abstract

A generalization of processing problems with restricted capacities is introduced. In a processing problem there is a finite set of jobs, each requiring a specific amount of effort to be completed, whose costs depend linearly on their completion times. The new aspect is that players have interest in all jobs. The corresponding cooperative game of this generalization is proved to be totally balanced.

Keywords: Processing games, scheduling, core allocation.

JEL Classification Number: C71.

1 Introduction

In a processing problem there is a finite set of jobs, each requiring a specific amount of effort to be completed, whose costs depend linearly on their completion times. There are no restrictions whatsoever on the processing schedule. The main feature of the model is a capacity restriction, i.e., there is a maximum amount of effort per time unit available for handling jobs.

In Meertens, Borm, Quant, and Reijnierse (2004) processing problems are studied in a cooperative game theory framework. They consider a model in which a player, endowed with an individual capacity for handling jobs, is assigned to exactly one job (i.e. there is a one-one correspondence between the jobs and the players). Each coalition of cooperating players in fact faces

¹Corresponding author. Email: quant@uvt.nl

²CentER and Department of Econometrics & OR, Tilburg University, P.O. Box 90153, 5000 LE Tilburg, The Netherlands.

³Department of Mathematics, University of Nijmegen, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands.

a processing problem with the coalitional capacity being the sum of the individual capacities of the members. The corresponding processing game summarizes the minimal joint costs for every coalition. They prove that these processing games are totally balanced by providing an explicit core element.

In this paper, we study a generalization of the model. This generalization considers processing situations with *shared interest*. In a processing situation with shared interest each player, still endowed with a personal capacity, may have interest for *each* job that has to be completed and the number of jobs need not to be equal to the number of players. More particularly, the cost-coefficients do *not* only depend on the jobs, but also on the players. Each player can have an interest for *every* job to be completed, this interest may be different for each player. This situation incorporates the previous model; a standard processing situation can be described by a processing situation with shared interest in which each player has exactly one strictly positive cost coefficient and so does each job. The related processing games with shared interest are proved to be totally balanced.

2 Processing games

A *processing problem* \mathcal{P} with restricted capacity consists of a tuple

$$\langle J, p = (p_j)_{j \in J}, \alpha = (\alpha_j)_{j \in J}, \beta \rangle.$$

Here, J is a finite set of jobs that need to be completed. The vector $p \in \mathbb{R}_+^J$ contains the *processing demands* or efforts of the jobs, furthermore $\alpha \in \mathbb{R}_+^J$ is the vector of *cost coefficients* and β is a strictly positive real number denoting the maximum available effort per time unit, or shortly *capacity*. The costs for job $j \in J$ to be uncompleted for a period of time t equals $\alpha_j \cdot t$. The objective is to find a feasible schedule such that the total costs are minimized. This minimum is denoted by $c(\mathcal{P})$. To attain these minimal costs, the jobs should be completed one after an other. It is proved that to complete all jobs such that total costs are minimized, it is optimal to complete the jobs one by one. The order in which this is done depends on the urgency of the jobs. This leads to the following proposition.

Proposition 2.1 (cf. Smith (1956)). Let \mathcal{P} be a processing problem such that $J = \{1, \dots, |J|\}$ and the jobs are numbered such that $\frac{\alpha_1}{p_1} \geq \dots \geq \frac{\alpha_{|J|}}{p_{|J|}}$.

Then it is optimal to process the jobs in increasing order and

$$c(\mathcal{P}) = \frac{1}{\beta} \sum_{i=1}^{|J|} \alpha_i \cdot [p_1 + \dots + p_i].$$

In a *processing situation* $\langle N, J, p = (p_j)_{j \in J}, \alpha = (\alpha_j)_{j \in J}, (\beta_i)_{i \in N} \rangle$ there is a finite set of *players* N , in which each player $i \in N$ is endowed with a strictly positive *capacity* of β_i , in order to perform jobs. Each *job* $j \in J$ has a *processing demand* p_j and cost coefficient α_j , both in \mathbb{R}_+ . As long as job j is uncompleted, it generates a cost of size α_j per time unit. Each player has to complete one specific job in J . Since each player has only one job and players have not the same job, there is a one-one correspondence between players and jobs and no confusion occurs when the processing demand and the cost coefficient of the job of player i are denoted by p_i and α_i respectively.

Let $S \subseteq N$ be a coalition of players who decide to cooperate. This coalition has the disposal of the individual capacities of all of its members, i.e. coalition S can maximally generate an amount of effort of size $\beta(S) := \sum_{i \in S} \beta_i$ per time unit. The aim of coalition S is to complete all jobs of its members, such that aggregate costs are minimized. This situation gives rise to the processing problem

$$\mathcal{P}(S) := \langle J(S), (p_i)_{i \in S}, (\alpha_i)_{i \in S}, \beta(S) \rangle,$$

in which $J(S)$ denotes the set of jobs of players in S . The *processing game* $\langle N, c^{\mathcal{P}} \rangle$ in which $c^{\mathcal{P}} : 2^N \rightarrow \mathbb{R}_+$ is the map defined by

$$c^{\mathcal{P}}(S) := c(\mathcal{P}(S)) \text{ for all } S \subseteq N.$$

Theorem 2.1 (cf. Meertens et al. (2004)). Processing games are totally balanced.

Let $\langle N, J, p = (p_j)_{j \in J}, \alpha = (\alpha_j)_{j \in J}, (\beta_i)_{i \in N} \rangle$ be a processing situation. In fact the authors show that the vector $y \in \mathbb{R}^N$ with

$$y_i = \frac{\alpha_i}{\beta(N)} \sum_{k=1}^i p_k + \tau_i - \frac{\beta_i}{\beta(N)} \sum_{k=1}^n \tau_k, \quad (1)$$

for all $i \in N$, is a core allocation of the corresponding processing game, provided that $N := \{1, \dots, n\}$ and $\frac{\alpha_1}{p_1} \geq \dots \geq \frac{\alpha_n}{p_n}$. Here τ_i denotes the tax paid by player i , and this tax is defined by:

$$\tau_i := \frac{p_i}{\beta(N)} \cdot \left[\frac{1}{2} \cdot \alpha_i + \sum_{k=i+1}^n \alpha_k \right].$$

So, the amount player $i \in N$ has to pay in the core allocation y consists of three parts. The first part $\frac{\alpha_i}{\beta(N)} \sum_{k=1}^i p_k$ gives the *actual costs* of player i . These are the product of the cost coefficient of player i and the completion time of his job. The second part τ_i is the tax paid by player i , which is paid because all players work on the job of player i and all jobs with lower urgencies have to wait for completion. The sum of taxes is redivided among the players proportionally to their capacities. This results in the third part $-\frac{\beta_i}{\beta(N)} \sum_{k=1}^n \tau_k$, called the *subsidy* to player i . It is done in order to reward players with large capacity. A more detailed explanation of this core element can be found in Meertens et al. (2004).

3 Processing games with shared interest

In this section we analyze an extension of processing situations and the corresponding processing games. In stead of each player having *one* job, we allow them to have interest in several jobs. We prove that the corresponding games are totally balanced.

A *processing situation with shared interest* P can be described by a tuple $\langle N, J, p = (p)_{j \in J}, A, \beta = (\beta_i)_{i \in N} \rangle$. Here, N is a finite set of players and J a finite set of jobs. The vector $p \in \mathbb{R}_+^J$ contains the processing demands of the jobs. β contains the capacities with which the players are endowed. The matrix $A \in \mathbb{R}_+^N \times \mathbb{R}_+^J$ contains the cost coefficients of all players for all possible jobs. The number A_{ij} denotes the cost coefficient of player $i \in N$ with respect to job $j \in J$. If $A_{ij} = 0$ then player i has no interest in job j . Contrary to the original setting it is now possible for a player to have interest in several jobs and a job can be of interest for more than one player. The original problem can be modeled as a processing situation with shared interest by choosing the matrix A to be a squared diagonal matrix. Let $i \in N$. The set of jobs in which player i is interested is denoted by $J_i = \{j \in J \mid A_{ij} > 0\}$.

If a coalition $S \subseteq N$ decides to cooperate, its members have a total capacity of $\beta(S) := \sum_{i \in S} \beta_i$ available in order to construct a schedule which completes all their jobs, such that total weighted costs are minimized. This situation gives rise to the processing problem

$$\mathcal{P}(S) := \langle J(S), (p_j)_{j \in J(S)}, (A_j(S))_{j \in J(S)}, \beta(S) \rangle,$$

in which $J(S)$ denotes $\cup_{i \in S} J_i$, the set of all jobs which are of interest for players of S and $A_j(S) = \sum_{i \in S} A_{ij}$ is the total cost coefficient of coalition S for job j . Analogous to the problem in which each player has only one

job, one can associate a *processing game with shared interest* $\langle N, c^P \rangle$ with $c^P : 2^N \rightarrow \mathbb{R}_+$ as follows:

$$c^P(S) := c(\mathcal{P}(S)) \text{ for all } S \subseteq N.$$

Example 3.1. Suppose there are two students who share an apartment. Since they just moved in, they need to buy some domestic appliances as a refrigerator, a stove and a washing machine. Each student has a restricted budget per month which is reserved for these purchases. Suppose student 1 has a budget of 100 euro a month and the budget of student 2 equals 200 euro a month. The costs of buying a new refrigerator, stove or washing machine are 300, 150 and 600 euro respectively. Both students have different priorities with respect to the three goods. For example student 1 can use the washing machine of a friend, so he is more in need of a refrigerator than a washing machine, on the other hand student 2 does not like cooking so he values a stove low. This situation can be modeled as a processing situation with shared interest, with player set $N = \{1, 2\}$, capacities $\beta = (100, 200)$ and processing demands $(300, 150, 600)$. The matrix of cost coefficients represents the priorities of the students with respect to the goods and reflects the dissatisfaction of a student per month if he cannot make use of a certain machine. Suppose the cost coefficient matrix equals

$$A = \begin{pmatrix} 3 & 2 & 1 \\ 4 & 1 & 5 \end{pmatrix}.$$

The first column denotes the priority of the players with respect to the refrigerator, the second and third column denote the priorities with respect to the stove and the washing machine. If player 1 only has his own budget, then he first buys a stove, then a refrigerator and finally a washing machine, since $\frac{2}{150} \geq \frac{3}{300} \geq \frac{1}{600}$. The total measure of "dissatisfaction" for student 1 equals $c^P(\{1\}) = 4\frac{1}{2} \cdot 3 + 1\frac{1}{2} \cdot 2 + 10\frac{1}{2} \cdot 1 = 27$. Similarly $c^P(\{2\}) = 1\frac{1}{2} \cdot 4 + 5\frac{1}{4} \cdot 1 + 4\frac{1}{2} \cdot 5 = 33\frac{1}{4}$. If the students buy the domestic appliances together, they first buy a refrigerator, then a stove and finally a washing machine, and $c^P(N) = 32\frac{1}{2}$. \circ

Theorem 3.1. A processing game with shared interest is balanced.

To prove Theorem 3.1, we first prove balancedness for *processing situations with multiple jobs*. In a processing situation with multiple jobs players can have interest in more than one job, but each job is still of interest for only one player. In fact each player "owns" a set of jobs.

The next lemma states that a multiple jobs processing game is (totally) balanced. The proof uses a processing game in the original setting.

Lemma 3.1. Every multiple jobs processing game is totally balanced.

Proof: Let (N, c^P) be a multiple jobs processing game with associated situation $P = \langle N, J, p, A, \beta \rangle$, i.e. $J_i \cap J_k = \emptyset$ for all $i, k \in N$ with $i \neq k$. Define the processing situation $\bar{P} = \langle J(N), J, p, \bar{\alpha}, \bar{\beta} \rangle$ as follows. For each job in J a player is created, so the player set of \bar{P} is $J(N)$ and $J(N) = J$. A 'player' j , which is actually a job, keeps its processing demand p_j and its cost coefficient becomes $\bar{\alpha}_j = A_{ij} > 0$, in which i is the unique player such that $A_{ij} > 0$. The capacity β_i of a regular player i in N is split equally over all new players originating from his job set J_i , resulting in:

$$\bar{\beta}_j := \frac{\beta_i}{|J_i|} \text{ for all } i \in N, j \in J_i.$$

It is left to the reader to verify that for each coalition $S \subseteq N$ the following is true,

$$c^P(S) = c^{\bar{P}}(J(S)). \quad (2)$$

According to Theorem 2.1, the game $\langle J(N), c^{\bar{P}} \rangle$ is totally balanced and has a core allocation y in $\mathbb{R}^{J(N)}$, see equation (1). Define x in \mathbb{R}^N by $x_i := \sum_{j \in J_i} y_j$ for all i in N . Since equation (2) is valid and $y \in \mathcal{C}(c^{\bar{P}})$, it follows that $x \in \mathcal{C}(c^P)$. \square

In order to prove that processing games with shared interest are balanced it is shown a core allocation exists. Before giving the proof an intuitive explanation of this core element is given. It is based on the allocation for processing games, see expression (1). Note that the tax τ_i depends only on the processing time and the cost coefficient of the job of player i , say job j . Hence it can be considered to be a property of job j and be called τ_j . In this view one can easily extend this allocation to an allocation for a processing game with shared interest. Let $P = \langle N, J, p, A, \beta \rangle$ be a processing situation with shared interest. Then the allocation x with for all $i \in N$

$$x_i := \left(\sum_{j \in J} \frac{A_{ij}}{\beta(N)} \sum_{k=1}^j p_k \right) + \left(\sum_{j \in J} \frac{A_{ij}}{A_j(N)} \tau_j \right) - \frac{\beta_i}{\beta(N)} \sum_{j \in J} \tau_j. \quad (3)$$

is in the core of the game $\langle N, c^P \rangle$, provided that $J = \{1, \dots, |J|\}$ and $\frac{A_{1(N)}}{p_1} \geq \dots \geq \frac{A_{|J|(N)}}{p_{|J|}}$ (this means that if the grand coalition cooperates,

the jobs are performed in the order $(1, \dots, |J|)$. The value τ_j , is for every $j \in J$ defined by

$$\tau_j = \frac{p_j}{\beta(N)} \left[\frac{1}{2} A_j(N) + \sum_{k=j+1}^{|J|} A_k(N) \right].$$

The division of the allocation into three parts is still accurate. The first part concerns the actual costs of the jobs. Each player i has to pay his personal actual costs of each job j . The second part gives the taxes of the jobs. These are imposed in proportion with the costs coefficients. The third part concerns the subsidies, which are still allocated proportionally with respect to the capacities of the players.

The core element x is independent on the optimal order chosen:

Lemma 3.2. Let $\langle N, J, p, A, \beta \rangle$ be a processing situation with shared interest. The core allocation x as given by (3) does not depend on the choice of which optimal order is used to process the jobs.

Proof: Equation (3) can be rewritten as

$$x_i := \sum_{j \in J} \frac{A_{ij}}{A_j(N)} \left(\frac{A_j(N)}{\beta(N)} \sum_{k=1}^j p_k + \tau_j \right) - \frac{\beta_i}{\beta(N)} \sum_{j \in J} \tau_j. \quad (4)$$

The result now follows from Meertens et al. (2004), since they prove that if another optimal order is used, the total amount of taxes paid does not change, and neither does the term $\frac{A_j(N)}{\beta(N)} \sum_{k=1}^j p_k + \tau_j$ for each $j \in J$. \square

We now turn to the proof of Theorem 3.1.

Proof of Theorem 3.1: Let $P = \langle N, J, p, A, \beta \rangle$ be a processing situation with shared interest. Without loss of generality we assume that $J = \{1, \dots, |J|\}$ and $\frac{A_1(N)}{p_1} \geq \dots \geq \frac{A_{|J|}(N)}{p_{|J|}}$. Let x be the allocation as described in equation (3). We first show that it is sufficient to prove that x is a core allocation in the case $|N| = 2$. To prove that x is a core allocation if $|N| = 2$ an induction argument on the number of jobs for which both players have a positive cost coefficient is used.

Suppose that $P = \langle N, J, p, A, \beta \rangle$ is a situation such that $x \notin C(c^P)$. Then there exists a coalition $S \subset N$ such that $x(S) = \sum_{i \in S} x_i > c^P(S)$. Define a two person processing situation \bar{P} with shared interest as follows: $\langle \bar{N} = \{1, 2\}, J, p, \bar{A}, \bar{\beta} \rangle$. The job set and the vector of processing demands remain unchanged. Player 1 can be seen as the player representing coalition S

and player 2 as the player representing coalition $N \setminus S$. The matrix of cost coefficients \bar{A} is defined as follows: $\bar{A}_{1j} = A_j(S)$ and $\bar{A}_{2j} = A_j(N \setminus S)$ for all $j \in J$. The vector of capacities $\bar{\beta}$ is given by $\bar{\beta} = (\beta(S), \beta(N \setminus S))$. Note that the coalitions N and \bar{N} perform exactly the same jobs and for each job $j \in J$, $A_j(N) = A_j(\bar{N})$, hence the optimal orders for both problems coincide. The same is true for the coalitions S and $\{1\}$, and $N \setminus S$ and $\{2\}$. Hence

$$c^P(N) = c^{\bar{P}}(\bar{N}), \quad c^P(S) = c^{\bar{P}}(\{1\}), \quad c^P(N \setminus S) = c^{\bar{P}}(\{2\}).$$

Let \bar{x} be the allocation corresponding to \bar{P} (as described in (3)). Since $\bar{A}_{1j} = \sum_{i \in S} A_{ij}$ and $\bar{A}_{2j} = \sum_{i \in N \setminus S} A_{ij}$, it follows from equation (4) that

$$\begin{aligned} x(S) &= \bar{x}_1, \\ x(N \setminus S) &= \bar{x}_2. \end{aligned}$$

This indicates that \bar{x} is not a core allocation of the game $\langle \bar{N}, c^{\bar{P}} \rangle$. Hence, if there is a situation in which x is not a core allocation, then there is also such a situation with only two players.

Assume that $|N| = 2$. Let $\ell(P)$ be the number of jobs for which both players have a strictly positive cost coefficient (i.e. $\ell(P) := \{j \in J \mid A_{1j} > 0 \text{ and } A_{2j} > 0\}$). We prove our statement by induction.

If $\ell(P) = 0$ we deal with a processing problem with multiple jobs. In this case, x coincides with the allocation \bar{x} that is shown to be in the core of the corresponding processing game with multiple jobs in the proof of Lemma 3.1.

Let $k \in \mathbb{N}$. Assume that for each two person processing game with shared interest for which $\ell(P)$ does not exceed k , x is a core element. Let $P = \langle N = \{1, 2\}, J, p, A, \beta \rangle$ be a processing situation such that $\ell(P)$ equals $k + 1$. We prove that $x \in C(c^P)$, which completes the induction argument. Let j be a job with shared interest. Without loss of generality we assume that the processing demand of job j equals the total cost coefficient: $p_j = A_j(N)$ (this is just a matter of scaling). Then the urgency of job j equals 1. Define another processing situation $\bar{P} = \langle N, \bar{J}, \bar{p}, \bar{A}, \beta \rangle$. \bar{P} arises from P by splitting j into two jobs with single interest, both having urgency 1, like j . This yields

$$\begin{aligned} \bar{J} &= \{1, \dots, j-1, j_a, j_b, j+1, \dots, |J|\}, \\ \bar{p} &= \{p_1, \dots, p_{j-1}, A_{1j}, A_{2j}, p_{j+1}, \dots, p_{|J|}\}, \\ \bar{A} &= \begin{pmatrix} A_{11} & \dots & A_{1(j-1)} & A_{1j} & 0 & A_{1(j+1)} & \dots & A_{1|J|} \\ A_{21} & \dots & A_{2(j-1)} & 0 & A_{2j} & A_{2(j+1)} & \dots & A_{2|J|} \end{pmatrix}. \end{aligned}$$

Let \bar{x} be the allocation defined by equation (3) that corresponds to \bar{P} . By the induction hypothesis it holds that $\bar{x} \in C(c^P)$. Hence $c^{\bar{P}}(\{i\}) - \bar{x}_i \geq 0$ for $i \in \{1, 2\}$. Note that it is sufficient to prove that $c^P(\{i\}) - x_i \geq c^{\bar{P}}(\{i\}) - \bar{x}_i$, since this guarantees that x is a core element. It is equivalent to show that

$$c^P(\{i\}) - c^{\bar{P}}(\{i\}) \geq x_i - \bar{x}_i. \quad (5)$$

In order to prove inequality (5), we prove two statements:

$$c^P(\{i\}) - c^{\bar{P}}(\{i\}) \geq \frac{A_{1j}A_{2j}}{\beta_i} \quad (6)$$

$$x_i - \bar{x}_i = \frac{A_{1j}A_{2j}}{2\beta(N)} \quad (7)$$

for $i \in \{1, 2\}$. Inequality (5) immediately follows from inequality (6) and equality (7) and the fact that $\beta_i < \beta(N)$.

Let $i = 1$, note that the job set of player 1 in the new situation can be written as $\bar{J}_1 = (J_1 \setminus \{j\}) \cup \{j_a\}$. Let σ be the optimal order of player 1 for the processing problem $\mathcal{P}(\{1\})$ and let $\bar{\sigma}$ be the same order as σ , but job j is replaced by j_a . This means that player 1 completes the jobs in the old order σ , but in stead of job j , job j_a is performed. Order $\bar{\sigma}$ is a possible order to solve the processing problem $\bar{\mathcal{P}}(\{1\})$. We compare the costs of σ with the costs of $\bar{\sigma}$. Since the processing demands of jobs j and j_a are not equal (contrary to the cost coefficients!), this yields a cost reduction of at least

$$\frac{1}{\beta_1}(p_j - p_{j_a}) \cdot A_{1j} = \frac{1}{\beta_1}(A_j(N) - A_{1j})A_{1j} = \frac{A_{1j}A_{2j}}{\beta_1}.$$

Note that the ready times of jobs beyond j become smaller in $\bar{\sigma}$. This yields an extra cost reduction. Hence the value found above is a lower bound for the cost reduction. It is possible that for player 1 another order becomes optimal when facing $\bar{\mathcal{P}}(\{1\})$. This leads to the following estimation:

$$c^{\bar{P}}(\{1\}) \leq c(\bar{\sigma}) \leq c^P(\{1\}) - \frac{A_{1j}A_{2j}}{\beta_1},$$

where $c(\bar{\sigma})$ denotes the costs if player $\{1\}$ performs his job in the order $\bar{\sigma}$. The same argument holds for player 2, which proves inequality (6).

Jobs j_a and j_b have equal urgency for coalition N . Since x does not change if jobs with the same urgency are switched (Lemma 3.2), we assume that the order corresponding to \bar{x} equals $(1, \dots, j-1, j_a, j_b, j+1, \dots, |J|)$. Let $\bar{\tau}$ be the vector of taxes corresponding with \bar{x} . Denote $\sum_{k>j} A_k(N)$ by s . It is easy to verify the following equations:

$$\begin{aligned} \bar{\tau}_k &= \tau_k \text{ for all } k \in \bar{J} \setminus \{j_a, j_b\}, \\ \bar{\tau}_{j_a} &= \frac{A_{1j}}{\beta(N)} \left(\frac{1}{2} \bar{A}_{j_a}(N) + \bar{A}_{j_b}(N) + s \right), \\ \bar{\tau}_{j_b} &= \frac{A_{2j}}{\beta(N)} \left(\frac{1}{2} \bar{A}_{j_b}(N) + s \right). \end{aligned}$$

Because $A_{1j} = \bar{A}_{j_a}(N)$, $A_{2j} = \bar{A}_{j_b}(N)$ and $p_j = A_{1j} + A_{2j}$, we have

$$\bar{\tau}_{j_a} + \bar{\tau}_{j_b} = \frac{1}{\beta(N)} \left(\frac{1}{2} (A_{1j} + A_{2j})^2 + (A_{1j} + A_{2j})s \right) = \frac{p_j}{\beta(N)} \left(\frac{1}{2} p_j + s \right) = \tau_j.$$

This implies that the total amount of taxes paid, does not change: $\sum_{j \in J} \bar{\tau}_j = \sum_{j \in J} \tau_j$. Since $\bar{p}_{j_a} + \bar{p}_{j_b} = p_j$, the completion times of all other jobs remain unchanged. We can now calculate the difference between the two allocations:

$$\begin{aligned} x_1 - \bar{x}_1 &= \frac{A_{1j}}{\beta(N)} \sum_{k=1}^j p_k + \frac{A_{1j}}{A_j(N)} \tau_j - \frac{\bar{A}_{1j_a}}{\beta(N)} \sum_{k=1}^{j_a} \bar{p}_k - \frac{\bar{A}_{1j_a}}{A_{j_a}(N)} \bar{\tau}_{j_a} \\ &= \frac{A_{1j}}{\beta(N)} p_j - \frac{A_{1j}}{\beta(N)} \bar{p}_{j_a} + \frac{A_{1j}}{A_j(N)} \tau_j - \bar{\tau}_{j_a} \\ &= \frac{A_{1j} A_{2j}}{\beta(N)} + \frac{A_{1j}}{\beta(N)} \left(\frac{1}{2} A_j(N) + s \right) - \frac{\bar{A}_{1j_a}}{\beta(N)} \left(\frac{1}{2} \bar{A}_{j_a}(N) + \bar{A}_{j_b}(N) + s \right) \\ &= \frac{A_{1j} A_{2j}}{\beta(N)} + \frac{A_{1j}}{\beta(N)} \cdot \frac{1}{2} (A_{1j} + A_{2j}) - \frac{A_{1j}}{\beta(N)} \left(\frac{1}{2} A_{1j} + A_{2j} \right) \\ &= \frac{A_{1j} A_{2j}}{2\beta(N)}. \end{aligned}$$

And

$$\begin{aligned} x_2 - \bar{x}_2 &= \frac{A_{2j}}{\beta(N)} \sum_{k=1}^j p_k + \frac{A_{2j}}{A_j(N)} \tau_j - \frac{\bar{A}_{2j_b}}{\beta(N)} \sum_{k=1}^{j_b} \bar{p}_k - \frac{\bar{A}_{2j_b}}{A_{j_b}(N)} \bar{\tau}_{j_b} \\ &= \frac{A_{2j}}{A_j(N)} \tau_j - \bar{\tau}_{j_b} \\ &= \frac{A_{2j}}{A_j(N)} \tau_j - \frac{A_j(N)}{A_j(N)} (\tau_j - \bar{\tau}_{j_a}) \\ &= \frac{-A_{1j}}{A_j(N)} \tau_j + \bar{\tau}_{j_a} \\ &= \frac{-A_{1j}}{\beta(N)} \left(\frac{1}{2} A_j(N) + s \right) + \frac{\bar{p}_{j_a}}{\beta(N)} \left(\frac{1}{2} \bar{A}_{j_a}(N) + \bar{A}_{j_b}(N) + s \right) \\ &= \frac{-A_{1j}}{\beta(N)} \cdot \frac{1}{2} (A_{1j} + A_{2j}) + \frac{A_{1j}}{\beta(N)} \left(\frac{1}{2} A_{1j} + A_{2j} \right) \\ &= \frac{A_{1j} A_{2j}}{2\beta(N)}. \end{aligned}$$

This proves equation (7) and the theorem. \square

References

- Meertens, M., P. Borm, M. Quant, and H. Reijnierse (2004). Processing games with restricted capacities. CentER DP 2004-83, Tilburg University, Tilburg, The Netherlands.
- Smith, W. (1956). Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, **3**, 59–66.