Equilibrium in a two-agent Assignment Problem

Giovanni Felici

Istituto di Analisi dei Sistemi ed Informatica del CNR,
Viale Manzoni, 30, Roma, Italy
E-mail: felici@iasi.cnr.it
Website: www.iasi.rm.cnr.it/iasi/personnel/felici.html

Mariagrazia Mecoli

Università degli Studi di Roma 'La Sapienza',
Roma, Italy
E-mail: mariagrazia.mecoli@uniroma1.it

Pitu B. Mirchandani

SIE, University of Arizona,
1127 E. James E. Rogers Way, Engineering Building 20,
P.O. Box 210020 Tucson, AZ, USA
E-mail: pitu@sie.arizona.edu
Website: www.sie.arizona.edu/faculty/mirchandani.html

Andrea Pacifici*

Università degli Studi di Roma di Tor Vergata,
Via del Politecnico, 1, Roma, Italy
E-mail: pacifici@disp.uniroma2.it
Website: www.disp.uniroma2.it/users/pacifici
*Corresponding author

Abstract: In this paper we address a particular generalisation of the Assignment Problem (AP) in a Multi-Agent setting, where distributed agents share common resources. We consider the problem of determining Pareto-optimal solutions that satisfy a fairness criterion (equilibrium). We show that the solution obtained is equivalent to a Kalai–Smorodinsky solution of a suitably defined bargaining problem and characterise the computational complexity of finding such an equilibrium. Additionally, we propose an exact solution algorithm based on a branch-and-bound scheme that exploits bounds obtained by suitably rounding the solutions of the corresponding linear relaxation, and give the results of extensive computational experiments.

Keywords: competitive assignment; equilibrium; Pareto-optimality.

Reference to this paper should be made as follows: Felici, G., Mecoli, M., Mirchandani, P.B. and Pacifici, A. (2009) 'Equilibrium in a two-agent Assignment Problem', *Int. J. Operational Research*, Vol. 6, No. 1, pp.4–26.

Biographical notes: Giovanni Felici is currently Tenured Researcher at the Isitituto di Analisi dei Sistemi ed Informatica of the Italian National Research Council, in Rome (Italy). His main area of expertise is operations research, subject in which he received his MSc at the University of Lancaster (UK) and his PhD at the University of Rome 'La Sapienza'. His research interests cover integer optimisation, polyhedral methods, optimisation algorithms for data mining and their applications. He recently approached the promising field of research related with the mathematical models and algorithms for the solution of multi-agents optimisation problems in a competitive environment.

Mariagrazia Mecoli obtained her PhD Degree in Operations Research from the University of Rome 'La Sapienza' in 2005. Her primary research interests are in combinatorial optimisation with applications in logistics, transportation, and resource assignment. She is currently working at the Department of Statistics and Probability at the University of Rome, 'La Sapienza'.

Pitu B. Mirchandani, Salt River Project Professor of Technology, Public Policy, and Markets, has joint appointments in the Department of Systems and Industrial Engineering and the Department of Electrical and Computer Engineering at the University of Arizona. He is the Director of the ATLAS Research Center. His expertise includes logistics, games and equilibria, design of decision/control systems, and their application in transportation and distribution. He received a Bachelor's Degree and a Master's Degree in Engineering from the University of California-Los Angeles, a Master's Degree in Aeronautics and Astronautics, and a Doctorate in Operations Research from the Massachusetts Institute of Technology.

Andrea Pacifici is an Assistant Professor in operations research at the Engineering Faculty of the University of Rome Tor Vergata. He received a bachelor's degree in Information Engineering and a PhD in operations research both at the University of Rome 'La Sapienza'. His research is mainly concerned with algorithms design and computational complexity characterisation for combinatorial optimisation problems with applications to scheduling, logistics, manufacturing, telecom and multi-agent systems.

1 Introduction

The standard *Assignment Problem* (AP) consists in finding a minimum cost matching of jobs to machines. It is a well known combinatorial optimisation problem solvable in polynomial time (Bertsekas, 1981; Burkard, 2002; Khun, 1955). A considerable amount of literature, inspired by many practical applications and operations research models, addresses various extensions of AP. Among the others, we mention the *Generalised Assignment Problem*, where machines have capacities and each job has a size (see for instance Benders and van Nunen, 1983; Fisher et al., 1986; Higgins, 2001; Ross and Soland, 1975; Savelsbergh, 1997, and in particular, Cattryse and Wassenhove, 1992, for a survey of algorithms), and the class of *Constrained Assignment Problems* in which one has to find an optimal assignment while satisfying various kinds of side constraints (Aggarwal, 1985; Aboudi and

Nemhauser, 1991; Gupta and Sharma, 1981; Mazzola and Neebe, 1986; Punnen and Aneja, 1995). From the computational complexity standpoint, unlike the standard problem, most of these extensions fall in the class of $\mathcal{NP}$-hard problems.

In this paper, we focus on a particular generalisation of AP in a *Multi-Agent* setting, where two agents have their own sets of jobs and want to assign them to a set of machines. Each agent wishes to minimise an objective function depending on the assignment costs of its own jobs only, thus competing for the usage of the shared machines. We refer to this class of problems as *Competitive Assignment Problem* (CAP).

Multi-agent systems have recently received an increasing amount of attention in the optimisation literature (Arbib and Rossi, 2000; Agnetis et al., 2004; Felici et al., 2004; Oliveira et al., 1999). These systems are comprised by distributed autonomous agents which share common resources. Since there is no central coordinator, the agents, in pursuing their objectives, may give raise to conflicts. If this is the case, in order to reach a compromise, the agents want to negotiate over a suitable subset of the feasible agreements (assignments) where a better assignment for one agent necessarily results in a worse assignment for the other. These kind of solutions are known as *non-dominated* or *Pareto-optimal* solutions. Throughout the paper we make use of the term Pareto-optimal or, briefly, PO solution.

Our problem also presents analogies with a special cooperative game with nontransferable utility, namely the two-person *bargaining problem*, originally introduced by Nash (1950). The classical axiomatic characterisations of bargaining solutions are based on the assumption that the set of feasible agreements is convex. In the literature, some studies address nonconvex bargaining problems, and present extensions of classical equilibrium solutions such as those by Nash and Kalai and Smorodinsky (1975). In this context, a number of alternative axiomatic characterisations and different solutions are given for the class of compact and comprehensive (nonconvex) set of agreements (Conley and Wilkie, 1991, 1996; Herrero, 1989; Lin, 1997) or for bargaining problems where the number of agreements is finite (Kaneko, 1980; Mariotti, 1998). However, our study is related to a particular class of finite, non-convex bargaining problem where the set of agreements is discrete.

CAP may indeed be regarded as a two person bargaining problem where the set of the feasible agreements is discrete (and therefore nonconvex). In this paper, rather than providing an axiomatic characterisation of a bargaining solution, we aim at developing numerical methods to obtain a solution with desirable properties. We therefore

- consider a generalisation of the Kalai–Smorodinsky solution, called *Equilibrium Assignment*
- formulate the problem of finding such a solution as an integer program, namely the *Equilibrium Assignment Problem* (EAP)
- characterise its computational complexity
- design and test an exact solution algorithm.

The paper is organised as follows. Section 2 introduces CAP and gives the main notation used throughout this paper. In Section 3, the definition of Pareto-optimal solution is given. Special PO solutions, called *efficient* solutions, are also introduced. Issues related to the problem of finding one or all the PO solutions are also discussed.

The definition of EAP, which is modelled as a $\{0,1\}$ integer program, is illustrated in Section 4. We show that this problem is $\mathcal{NP}$-hard, and prove that an optimal solution of the linear relaxation of EAP is a Kalai–Smorodinsky solution. The latter solution is used to obtain primal (i.e., feasible solution values) and dual (i.e., lower) bounds that one may exploit in an exact algorithm to solve EAP. Such an algorithm, based on a branch-and-bound scheme, is illustrated together with rounding techniques devised to obtain the primal (upper) bound for EAP. Sections 5 and 6 present computational experiences: design of the experiments, solver tools and alternative strategies are discussed. Finally, conclusions are drawn and directions for future research are given in Section 7.

2 Problem statement

The standard AP consists in finding a minimum cost matching of n jobs with n machines, and can be formulated as follows:

$$\begin{aligned} \min \; & \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\ & x \in \mathcal{A} \\ & x \in \{0,1\}^{n\times n} \end{aligned} \tag{1}$$

where $c_{ij} \in \mathbb{Z}_+$ is the cost of assigning job i to machine j; $\mathcal{A}$ is the *assignment polytope*, $\mathcal{A} = \left\{x \in \mathbb{R}^{n\times n} : \sum_{i=1}^{n} x_{ij} = 1, \sum_{j=1}^{n} x_{ij} = 1, x \geq 0_{n\times n}\right\}$; the binary variable x_{ij} is equal to 1 if job i is assigned to machine j and 0 otherwise.

Several methods for solving AP in polynomial time are known: some use linear programming techniques, exploiting the integrality of the assignment polytope (see Khun, 1955), while others use different combinatorial algorithms (see Bertsekas, 1981; Burkard and Cela, 1999; Gabow and Tarjan, 1989).

In this paper we consider two competing agents, referred to as Agent A and B, owning two disjoint sets of jobs to be assigned to a set M of n machines. Agent A is identified by the set of its jobs $I_A = \{a_1, a_2, \ldots, a_{n_A}\}$, while Agent B by the set of its jobs $I_B = \{b_1, b_2, \ldots, b_{n_B}\}$, with $I_A \cap I_B = \emptyset$, and (with no loss of generality) $n_A + n_B = n$.

The set of feasible solutions for CAP coincides with the set of feasible solutions for AP, i.e., formally, with the set of vertices $\mathrm{Ext}(\mathcal{A})$ of the assignment polytope $\mathcal{A}$.

We write an assignment x as $x = (x_A, x_B)$ if we are distinguishing the vectors' components corresponding to the jobs belonging to the two agents. Given a feasible assignment x, we denote by $c_A(x)$, $c_B(x)$ the assignment costs for Agent A and B, respectively.

Let us call P^A the AP of Agent A, that is the problem of assigning its n_A jobs to the n machines minimising $c_A(x)$, and P^B the corresponding problem for Agent B. Both problems P^A and P^B are polynomially solvable as they are standard APs.

Let M_A^*, and M_B^*, be the set of machines used in the optimal solution of problem P^A, and P^B, respectively. If $M_A^* \cap M_B^* = \emptyset$, then there is no conflict in the usage of the machines, and there is a unique trivial PO solution. On the contrary, the example below illustrates a case where conflicts arise on three machines.

Example 1: Consider the following instance of CAP, where $n_A = n_B = 5$, $M = \{1, 2, \ldots, 10\}$, and the costs for the two agents are given by the following two matrices:

$$\{c_{ij}^A\} = \begin{bmatrix} 2 & 3 & 4 & 5 & 4 & 3 & 2 & 4 & 7 & 7 \\ 3 & 5 & 3 & 4 & 6 & 4 & 2 & 5 & 6 & 6 \\ 3 & 4 & 3 & 3 & 5 & 8 & 5 & 5 & 5 & 8 \\ 5 & 6 & 3 & 2 & 6 & 6 & 6 & 1 & 4 & 7 \\ 3 & 4 & 4 & 5 & 6 & 6 & 6 & 7 & 8 & 6 \end{bmatrix} \qquad \{c_{ij}^B\} = \begin{bmatrix} 3 & 3 & 3 & 4 & 3 & 5 & 2 & 5 & 6 & 6 \\ 5 & 5 & 8 & 2 & 2 & 5 & 2 & 2 & 5 & 8 \\ 4 & 6 & 3 & 7 & 4 & 4 & 2 & 4 & 7 & 7 \\ 6 & 4 & 8 & 2 & 3 & 3 & 4 & 3 & 5 & 4 \\ 3 & 3 & 5 & 4 & 6 & 4 & 7 & 5 & 5 & 6 \end{bmatrix}.$$

The optimal solution of P^A is $\{(a_1, 6), (a_2, 7), (a_3, 4), (a_4, 8), (a_5, 1)\}$, where (a_i, j) indicates the matched pair, $i = 1, \ldots, n_A$, $j \in M$, and its value is 12. Analogously, the optimal assignment of Agent B (solution of P^B) is $\{(b_1, 2), (b_2, 5), (b_3, 7), (b_4, 4), (b_5, 1)\}$ with optimal value equal to 12. Then, $M_A^* = \{1, 4, 6, 7, 8\}$, and $M_B^* = \{1, 2, 4, 5, 7\}$. Hence, in this case, there is a conflict on machines 1, 4, and 7.

3 Pareto optimality

When the intersection between M_A^* and M_B^* is not empty, as in the Example 1, then it makes sense to look for Pareto-optimal solutions, which are defined as follows.

A feasible assignment x is Pareto-optimal if there is no other feasible assignment x', with $x' \neq x$, such that:

$$c_A(x') < c_A(x), \;\; c_B(x') \leq c_B(x) \quad \text{or} \quad c_B(x') < c_B(x), \;\; c_A(x') \leq c_A(x).$$

On the other hand, if such an assignment x' exists, then we say that x is *dominated* and that x' *dominates* x.

We represent geometrically Pareto-optimal solutions in the *cost space*: given a feasible assignment $x \in \mathcal{A}$ we draw the point $(c_A(x), c_B(x))$ in the 2-dimensional euclidean space. Hereafter, unless the context is not clear, we indicate both a PO assignment x and its image in the cost space with the term ‘Pareto Optimal solution’. We refer to the set $\{(c_A(x), c_B(x)) : x \text{ is a PO solution}\}$ as the *Pareto-optimal frontier*.

3.1 Extreme PO solutions

Unless we are in the trivial case, at least two PO solutions may be obtained with an easy procedure. First, observe that, for any PO solution $x, c_A(x) \geq c_A^*$ and $c_B(x) \geq c_B^*$, where c_A^* and c_B^* are the optimal solution values of P^A and P^B. Suppose now that the optimal solution x_A^* of P^A is unique and solve the problem of assigning the jobs of I_B to the machines in $M \backslash M_A^*$. Call $c_{B|A}$ its optimal value. Then the point $(c_A^*, c_{B|A})$ is a PO solution; in fact, since P^A has a unique optimum, any possible improvement for Agent B results in an increased cost for Agent A.

In general, the optimal assignment for one agent may not be unique. When two or more equivalent optimal solutions are available for one agent, we will assume that this agent always picks the solution leaving the best possible set of machines for the other agent. This makes the solution obtained Pareto-optimal. Suppose we want an optimal assignment for Agent A (so that it obtains a cost c_A^*) simultaneously ensuring that the best possible machine set for Agent B is left. In other words, we are looking for an optimal solution $\tilde{x}$ of the following program: $\min\{c_B(x) : c_A(x) = c_A^*, x \in \mathcal{A} \cap \{0,1\}^{n\times n}\}$. Indeed, we may find $\tilde{x}$ by solving another standard AP where the objective is a suitable linear combination of the agents' cost functions: $Kc_A(x) + c_B(x)$. Factor K should be large enough so that no decrease in costs from the assignment of Agent B could drive the optimal solution toward a value that is no longer optimal for A. (Given that the cost coefficients are integer, we may choose, e.g., $K > UB^B - LB^B$ where UB^B and LB^B are upper and lower bounds for $c_B(x)$.)

In conclusion, the pair $(c_A^*, c_{B|A})$ equals $(c_A(\tilde{x}), c_B(\tilde{x}))$, and the corresponding solution $\tilde{x} = (x_A^*, x_{B|A})$ is Pareto-optimal. Symmetrically, the pair $(c_{A|B}, c_B^*)$, corresponding to solution $(x_{A|B}, x_B^*)$, is Pareto-optimal. Therefore the following observation holds.

Observation 2: Any PO solution x of an instance with costs $c^A(x)$ and $c^B(x)$ for Agent A and B respectively, is such that $c_A^* \leq c^A(x) \leq c_{A|B}$ and $c_B^* \leq c^B(x) \leq c_{B|A}$.

Note that, in the special case when $M_A^* \cap M_B^* = \emptyset$ the problem admits a unique Pareto-optimal solution x^* and the two points $(c_A^*, c_{B|A})$ and $(c_{A|B}, c_B^*)$ coincide. If this is not the case, the two PO points $(c_A^*, c_{B|A})$, and $(c_{A|B}, c_B^*)$ are the two extreme points of the Pareto frontier.

Example 3: Referring to the instance of CAP given in Example 1, we have $x_A^* = \{(a_1,6),(a_2,7),(a_3,4),(a_4,8),(a_5,1)\}$, and $x_B^* = \{(b_1,2),(b_2,5),(b_3,7),(b_4,4),(b_5,1)\}$ and the optimal values c_A^* and c_B^* are both equal to 12.

By eliminating the set of machines optimally assigned to Agent B, M_B, from the set of machines available to Agent A, the conditioned assignment of Agent A, $P^{A|B}$, becomes the problem of assigning the I_A jobs to the set of machines $M \backslash M_B = \{3,6,8,9,10\}$. The solution to this standard AP is $x_{A|B} = \{(a_1,6),(a_2,3),(a_3,9),(a_4,8),(a_5,10)\}$, with value $c_{A|B} = 18$. For Agent B, by solving $P^{B|A}$ we obtain a conditioned value $c_{B|A} = 17$ and the solution is $x_{B|A} = \{(b_1,2),(b_2,5),(b_3,3),(b_4,10),(b_5,9)\}$. In this case we have $c_A^* \neq c_{A|B}$ and $c_B^* \neq c_{B|A}$ and the corresponding two extreme points of the Pareto frontier are represented by (12, 17) and (18, 12). The other points composing the PO set are given by (13, 16), (15, 15), (16, 14), (17, 13); Figure 1 shows the values of the solutions in the PO set.

3.2 Number of PO solutions

Here we show that the number of Pareto-optimal assignments may grow exponentially with the size of the instance and therefore that, in general, it is not possible to generate all the PO points efficiently.

Proposition 4: *The number of Pareto-optimal solutions of a CAP is not polynomially bounded.*

Proof: To prove the proposition we build an instance where the number of PO solutions is exponential in the number of jobs. Assume that $n_A = n_B = \frac{n}{2}$ (n even) and that, for $1 \leq i \leq \frac{n}{2}, 1 \leq j \leq n, c_{ij}^A = c_{ij}^B = c_j = 2^{j-1}$, that is, the cost of assigning job i to a machine j depends only on the machine j.

It is straightforward to see that, given two different feasible assignment x and $x' \neq x$ in $\mathcal{A} \cap \{0,1\}^{n \times n}$, the following properties hold:

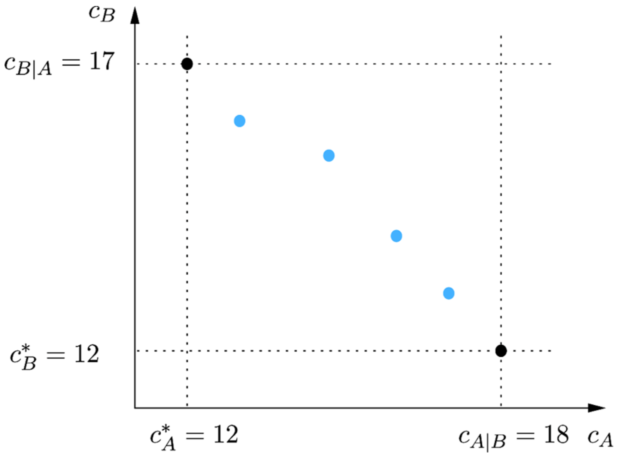
$$c_A(x) + c_B(x) = \sum_{j=1}^{n} c_j = 2^n - 1; \tag{2}$$

$$c_A(x) \neq c_A(x'). \tag{3}$$

For a given assignment x, let $M_A(x)$ denote the set of machines which the jobs of Agent A are assigned to. From the above properties, for any two assignments x and x' such that $M_A(x) \neq M_A(x')$, $c_A(x) \neq c_A(x')$. Moreover, if $c_A(x) > c_A(x')$ then $c_B(x) < c_B(x')$ and, conversely, if $c_A(x) < c_A(x')$ then $c_B(x) > c_B(x')$. Hence, any possible assignment for this instance is Pareto-optimal and there are as many PO pairs as the number of different possible subsets $M_A(x)$, i.e., $\binom{n}{\frac{n}{2}} = \Omega(\frac{2^n}{n})$ PO solutions.

□

Figure 1 The PO frontier of Example 3 (see online version for colours)



3.3 Efficient PO points

For the developments to follow, we define a special subset of PO points in the cost space, those lying on the convex hull of the Pareto frontier, that we call *efficient points*. For instance, the extreme points of the Pareto frontier, given by the two points $(c_A^*, c_{B|A})$ and $(c_{A|B}, c_B^*)$ are efficient. In Figure 1, the two extreme PO points (12, 17) and (18, 12) and the PO points (13, 16), (17, 13) are efficient, whereas (15, 15) and (16, 14) are not.

Differently from other PO solutions, any efficient point can be found by solving the following linear program for a given value of $\lambda, 0 \leq \lambda \leq 1$:

$$\begin{aligned} z(\lambda) = \min\ & \lambda \sum_{i \in I_A} \sum_{j \in M} c_{ij}^A x_{ij} + (1-\lambda) \sum_{i \in I_B} \sum_{j \in M} c_{ij}^B x_{ij} \\ & x \in \mathcal{A} \\ & x \in \{0,1\}^{n \times n}. \end{aligned} \tag{4}$$

Due to this property it is possible to verify in polynomial time if a feasible assignment is efficient (see Felici et al., 2004). However, it is important to observe that the number of efficient points is not polynomially bounded (in fact, in the proof of Proposition 4, all the PO points are efficient.)

3.4 Finding a PO solution

A possible approach (see Agnetis et al., 2004) to determine a PO assignment for CAP relies on the solution of a suitable integer program.

Given the sets I_A, I_B, the two objective functions $c_A(\cdot), c_B(\cdot)$, and an integer Q such that $c_B^* \leq Q \leq c_{B|A}$, we consider the problem of finding an assignment x^* such that Agent A gets the best assignment given that Agent B will accept an assignment of cost up to Q. We refer to this problem as *Constrained Problem* (CP). CP can be formulated as an integer linear program:

$$\begin{aligned} z_A(Q) = \min\ & c_A(x) \\ & x \in \mathcal{A} \\ & c_B(x) \leq Q \\ & x \in \{0,1\}^{n \times n}. \end{aligned} \tag{5}$$

It is easy to show that it is always possible to obtain a PO solution $(\bar{c}_A, \bar{c}_B)$ with

$$\begin{aligned} &\bar{c}_A \leq z_A(Q) \\ &\bar{c}_B \leq Q \end{aligned}$$

by means of a binary search on the Q values in the interval $[c_B^*, Q]$.

Observation 5: If $\widehat{A}$ is a solution algorithm for CP that runs in $O(f(n))$ time, then it is possible to use $\widehat{A}$ in a binary search framework to obtain a PO solution in $O(f(n) \cdot \log(c_{B|A} - c_B^*))$ time.

The following Proposition 6 shows that CP is a $\mathcal{NP}$*-hard* problem. Hence using CP to determine all the PO solutions, even when they are polynomially many, turns out to be a non-polynomial procedure.

Proposition 6: *CP is* $\mathcal{NP}$*-Hard.*

Proof: We prove that CP, as formulated in equation (5), is binary $\mathcal{NP}$-hard by making use of a reduction from the well known PARTITION problem: given n nonnegative integers $w_1, w_2, \ldots, w_n$ with $\sum_{i=1}^{n} w_i = W$, find, if it exists, a subset $S \subseteq \{1, \ldots, n\}$ such that $\sum_{i \in S} w_i = \frac{1}{2}W$. PARTITION remains $\mathcal{NP}$-hard if we restrict S to have cardinality equal to $\frac{n}{2}$ (see, e.g., Garey and Johnson, 1979).

We consider the following instance of CP where the cost c_{ij} of assigning a job i to a machine j is given by w_j for every $i \in I_A \cup I_B$, and $j \in M$. The number of jobs owned by Agent A is $n_A = \frac{n}{2}$; $M = \{1, \ldots, n\}$. Clearly $c_A^* = c_B^*$ since they are both equal to the sum of the $\frac{n}{2}$ lowest w_j's; while $c_{A|B} = c_{B|A}$ is equal to the sum of the $\frac{n}{2}$ highest w_j's.

The decision form of CP is the following: given two nonnegative real numbers $\bar{c}$ and Q, find if there is a solution $x \in \mathcal{A} \cap \{0,1\}^{n\times n}$, such that $c_A(x) \leq \bar{c}$ and $c_B(x) \leq Q$. Note that $c_A(x) + c_B(x) = W$. If we choose $\bar{c} = Q = W/2$, the decision problem is equivalent to the problem of determining a subset $S \subseteq M$ such that $|S| = \frac{n}{2}$ and $\sum_{j\in S} w_j = \frac{W}{2}$. This is equivalent to the restricted PARTITION problem defined above. □

4 Equilibrium between agents

In this section we focus on the problem of determining a special PO solution that provides an *equilibrium* of the system.

We are showing that this problem can be seen as a special bargaining problem with two players and that the optimal solution of a suitably defined linear program turns out to be a Kalai-Smorodinski solution of such a bargaining problem which provides bounds to the equilibrium we are looking for. Below we provide a formal description and a formulation for the problem of finding such an equilibrium. Also, we analyse whether an equilibrium always exists and/or it is unique.

Recall that the best and the worst possible PO alternative for one agent, say Agent A, are c_A^* and $c_{A|B}$ (respectively, c_B^* and $c_{B|A}$, for Agent B). Given an assignment x, we define the *ratio* of Agent A, $r_A(x)$ (of Agent B, $r_B(x)$) as:

$$r_A(x) = \frac{c_A(x) - c_A^*}{c_{A|B} - c_A^*} \quad \left(r_B(x) = \frac{c_B(x) - c_B^*}{c_{B|A} - c_B^*} \right). \tag{6}$$

These ratios are normalised distances of the agents' costs from their respective optimal values. Our definition of equilibrium solution exploits these ratios: the more similar the ratio values of a PO solution x are, the better is the compromise. In particular, if a PO solution x exists with $r_A(x) = r_B(x)$, then x represents a best compromise. In the cost space, the ideal points where $r_A(x) = r_B(x)$ define a line that we call *equilibrium ray*. In conclusion, we define the *Equilibrium Assignment Problem* (EAP) as follows.

$$r^* = \min\{\max\{r_A(x), r_B(x)\} : x \in \mathcal{A}, x \in \{0,1\}^{n\times n}\}. \tag{7}$$

To solve EAP, we may equivalently address two different integer linear problems. With reference to the cost space representation, the idea is to determine the two closest PO solutions to the equilibrium ray. This is modelled by the following integer program for the minimisation of r_A:

$$\begin{aligned} r_A^* = \min\ & r_A(x) \\ & x \in \mathcal{A} \\ & r_A(x) \geq r_B(x) \\ & x \in \{0,1\}^{n\times n}. \end{aligned} \tag{8}$$

Symmetrically, we obtain the second linear program for minimising r_B. Clearly, $r^* = \min\{r_A^*, r_B^*\}$.

In the following we give some properties of the optimal solutions of Problem 8.

4.1 Computational complexity issues

In what follows we will study Problem 8. We show below that EAP is $\mathcal{NP}$-Hard.

Proposition 7: *The Equilibrium Assignment Problem is $\mathcal{NP}$-Hard.*

Proof: We follow the same approach as in the proof of Proposition 6. For the same instance used in that proposition, the decision form of EAP, as defined in equation (8), is the following: given $\bar{c} \in \mathbb{R}_+$, find if there is a solution $x \in \mathcal{A} \cap \{0,1\}^{n\times n}$, such that $c_A(x) \geq c_B(x)$ and $c_A(x) \leq \bar{c}$. By definition, $c_A(x) = W - c_B(x)$, thus $c_A(x) \geq c_B(x)$ can be rewritten as $c_A(x) \geq \frac{W}{2}$. Let us choose $\bar{c} = W/2$; the decision problem is thus equivalent to the problem of determining a subset $S \subseteq M$ such that $\sum_{j\in S} w_j = \frac{W}{2}$ and this is equivalent to the restricted PARTITION problem defined in the proof of Proposition 6. □

The complexity of the problem implies that a polynomial exact solution algorithm is unlikely to exist.

In the following sections we first analyse a linear relaxation of EAP and develop a solution method based on branch-and-bound where a lower bound obtained starting from the solution of the linear relaxation is exploited.

4.2 Linear relaxation of EAP

Consider now the following problem (LEAP), which is a linear relaxation of EAP.

$$\begin{aligned} r^{LP} = \min\ & r_A(x) \\ & x \in \mathcal{A} \\ & r_A(x) \geq r_B(x) \\ & 0 \leq x_{ij} \leq 1 \quad i \in I_A \cup I_B,\ j \in M. \end{aligned} \tag{9}$$

In general, the solution value r^{LP} of LEAP is a lower bound for EAP. We denote by $\mathcal{A}'$ the non-empty polytope defined by its feasible solutions: $\mathcal{A}' = \mathcal{A} \cap R^{\geq}$, where $R^{\geq} = \{x \in \mathbb{R}^{n\times n} : r_A(x) \geq r_B(x)\}$. For the developments to follow, let also $R^{=} = \{x \in \mathbb{R}^{n\times n} : r_A(x) = r_B(x)\}$.

In the following we characterise the properties of the optimal solutions of Problem 9.

Proposition 8: *Let x^* be an optimal solution of Problem 9. Then $r_A(x^*) = r_B(x^*)$.*

Proof: We first rule out the trivial case when $c_A(x^*) = c^*_A$. Clearly, we have $r_A(x^*) = r_B(x^*) = 0$ and x^* is the unique PO solution.

Let x^* be an optimal solution of Problem (9) for which $c_A(x^*) > c^*_A$ (i.e., we have more than one PO solution), and suppose, by contradiction, that the proposition does not hold, that is $r_A(x^*) > r_B(x^*)$.

Since Problem (9) is a linear program, then $x^* \in \text{Ext}(\mathcal{A} \cap R^{\geq})$, that is, x^* is a vertex of the the polytope that defines the feasible region of Problem (9). Given that $x^* \in \text{Ext}(\mathcal{A} \cap R^{=})$ contradicts $r_A(x^*) > r_B(x^*)$, it must be $x^* \in \text{Ext}(\mathcal{A})$. Hence x^* is an integer feasible assignment.

Let $\tilde{x}$ be an assignment corresponding to $(c_A^*, c_{B|A})$ in the (c_A, c_B) plane. We have $r_A(\tilde{x}) \leq r_A(x^*)$ and $\bar{x} \neq x^*$. Consider a convex combination of x^* and $\tilde{x}, x = \lambda x^* + (1-\lambda)\tilde{x}$, with $0 \leq \lambda \leq 1$. Clearly $x \in \mathcal{A}$ and, for small enough values of $\lambda, r_A(x) \geq r_B(x)$. In conclusion, as $c_A^* < c_A(x) < c_A(x^*), x$ is a feasible solution of Problem (9) with $r_A(x) < r_A(x^*)$, thus, x^* is not optimal and this is a contradiction. □

By very similar arguments one can show that no feasible assignment may improve the cost for Agent B without increasing cost for Agent A and thus the following proposition holds.

Proposition 9: *Let x^* be an optimal solution of LEAP (9). Then no feasible assignment x exists such that $c_A(x) \leq c_A(x^*)$ and $c_B(x) < c_B(x^*)$, or $c_A(x) < c_A(x^*)$ and $c_B(x) \leq c_B(x^*)$. In the special case when x^* is integer then it is Pareto-optimal.*

We will now characterise the set $\mathrm{Ext}(\mathcal{A}')$ of the vertices of the polytope defined by the feasible solutions of LEAP.

We first recall some general concepts of polyhedral theory as illustrated in Nemhauser and Wolsey (1988). A *polyhedron* is the intersection of a finite set of half spaces; let $P = \{x \in \mathbb{R}^n; Ax \leq b\}$ be a polyhedron, where (A, b) is an $m \times (n+1)$ matrix. Let $(A^=, b^=)$ be the submatrix corresponding to the rows $i \in \{1, \ldots, m\}$, such that $a^i x = b_i$ for all $x \in P$. Given $\bar{x} \in P$, let $A_{\bar{x}}^=$ be the submatrix corresponding to the rows $i \in \{1, \ldots, m\}$, such that $a^i \bar{x} = b_i$.

The *dimension* of P is $\dim(P) = n - \mathrm{rank}(A^=)$. Below we will assume that the polyhedron P is *full-dimensional*, that is $\dim(P) = n$. Given $P \neq \emptyset$, and an inequality $\pi x \leq \pi_\circ$ satisfied for all $x \in P$, then the polyhedron $F = \{x \in P : \pi x = \pi_\circ\}$ is called *face* of P. An extreme point x of P is a 0-*dimensional* face of P, which means that, given that $\dim(P) = n$, $\mathrm{rank}(A_x^=) = n$, and there are n linear independent constraints that are tight at x. An edge of a polyhedron P is a 1-*dimensional* face of P, and it is given by the set of points x for which $\mathrm{rank}(A_x^=) = n - 1$.

Proposition 10: *Consider the full-dimensional polyhedron $P = \{x \in \mathbb{R}^n; Ax \leq b\}$, the half-space $H = \{x \in \mathbb{R}^n, dx \leq c\}$ and the polyhedron $P' = P \cap H$, with $P' \neq \emptyset$. Then, either P' is a face of P or* $\dim(P) = \dim(P')$.

Proof: It follows trivially from $\dim(P) = n - \mathrm{rank}(A^=)$, and $P' \neq \emptyset$. □

Proposition 11: *Consider the full-dimensional polyhedron $P = \{x \in \mathbb{R}^n; Ax \leq b\}$, the half-space $H = \{x \in \mathbb{R}^n, dx \leq c\}$ and the polyhedron $P' = P \cap H$, with $P' \neq \emptyset$. Then, any extreme point $x^* \in P'$ lies on an edge of P.*

Proof: First we consider the case in which P' is a face of P. In such case x^* is a vertex of P and the Proposition follows.

If P' is not a face of P, consider two sub-cases:

a $P' = P \cap H = P$; then the statement of the proposition is trivially satisfied since any extreme point $x \in P'$ is an extreme point of P, which lies on an edge of P.

b $P' = P \cap H \subset P$; then from Proposition 10 $\dim(P) = \dim(P') = n$, since P' is not a face of P. One of the two following possibilities must hold:

i $dx^* < c$. Given that x^* is extreme point of P', and that $\dim(P') = n$, then x^* satisfies at equality the n constraints. Therefore it must be $Ax^* = b$. It follows that x^* is an extreme point of P.

ii $dx^* = c$. As above, x^* satisfies at equality exactly $(n-1)$ constraints belonging to P. This trivially implies that x^* lies on an edge of P. □

From the above polyhedral considerations, the following results hold.

Proposition 12: *Given $\mathcal{A}$, the assignment polytope, and $\mathcal{A}'$, the polytope described by the feasible solutions of LEAP, for any vertex x^* of $\mathcal{A}'$ we have:*

i *x^* can be written as $x^* = fx' + (1-f)x'', 0 \leq f \leq 1$, where x' and x'' are two adjacent vertices of $\mathcal{A}$, with $x' \in \mathcal{A}'$ and $x'' \notin \mathcal{A}$*

ii *Let $x^* = \{x^*_{ij}\}_{\substack{i=1,\dots,n \\ j=1,\dots,n}}$. Then x^*_{ij} has at most two non-zeroes for each row i and column j.*

Proof: (i) Proposition 11 holds, thus x^* lies on an edge of $\mathcal{A}$. By definition, every point on an edge can be written as linear combination of its extremes. Thus, given $x', x'', x' \neq x''$, extreme points of $\mathcal{A}$, we have $x^* = fx' + (1-f)x'', f > 0$. Since x^* is an extreme point of $\mathcal{A}'$, then both x', x'' cannot be elements of $\mathcal{A}'$.

(ii) This result follows from the matrix structure of feasible assignments x' and x'', since any linear combination of two assignment solutions cannot have more than two positive elements for each row and column. □

4.3 LEAP and Kalai–Smorodinsky solutions

In this subsection we briefly provide an interpretation of the optimal solution of LEAP in the context of classical game theory.

We refer to the definition of (convex) bargaining problem given by Nash (1950). Here, two agents aim to maximise their utilities in bargaining, measured as numerical payoffs. The problem consists of a pair (F, d) where F is a nonempty, compact, and convex set of *feasible agreements*, and $d \in F$ is the *disagreement point*, i.e., the worst possible payoff pair. A *bargaining solution* $\phi(F, d)$ is a *rule* that assigns a feasible agreement to each bargaining problem. Nash defined a list of five *axioms* that a bargaining solution ought to satisfy ((i) Strong efficiency, (ii) Individual rationality, (iii) Scale covariance, (iv) Symmetry, and (v) Independence of irrelevant alternatives), and he showed that there exists a unique solution satisfying the five axioms. This solution is usually referred to as *Nash bargaining solution* of problem (F, d) (see Nash, 1950 for the details).

Starting from that seminal work, various axiomatic characterisation of a bargaining solution have been proposed in the game theory literature. In particular, due to many criticisms about the suitability of the Axiom (v), Kalai and Smorodinsky, later in 1975, replaced it with the (vi) Individual monotonicity axiom (see Kalai and Smorodinsky, 1975). Kalai and Smorodinsky proved that there is a unique solution satisfying axioms $(i), \ldots, (iv), (vi)$, and this is

$$\arg\max\left\{x_1 : \frac{x_1 - d_1}{u_1 - d_1} = \frac{x_2 - d_2}{u_2 - d_2};\ x \in F\right\}$$

i.e., a particular PO solution lying on the segment whose extrema are the disagreement point d and the ideal point u. According to many authors, a Kalai–Smorodinsky solution "seems to be a more solid candidate for a general principle for the choice of outcome" (Hougaard and Tvede, 2003).

We now introduce a two-person bargaining problem $(F_{\mathrm{LEAP}}, d_{\mathrm{LEAP}})$ associated to a given instance of LEAP, where the payoff function is given in terms of the cost functions of LEAP. For any (possibly fractional) assignment x, solution of LEAP, the payoff of Agent A is $f_A(x) = c_{A|B} - c_A(x)$ and, similarly, that of Agent B is $f_B(x) = c_{B|A} - c_B(x)$. Each agent wishes to maximise her own payoff. Now, the set of feasible payoffs for the agents is given by the set of points $F_{\mathrm{LEAP}} = \{(f_A(x), f_B(x)) : x \in \mathcal{A}\}$. Let the disagreement point be $d_{\mathrm{LEAP}} = (0, 0)$, and consider the ideal point u_{LEAP} given by $u_{\mathrm{LEAP}} = (c_{A|B} - c_A^*, c_{B|A} - c_B^*)$. Such situation can be also interpreted as the problem where two players, A and B, bargain over the set of feasible assignments $\mathcal{A}$ in order to minimise each cost function, i.e. $c_A(x)$ or $c_B(x)$. In this case the disagreement point corresponds to the worst assignment they both can get, given the presence of the other player, that is $d = (c_{A|B}, c_{B|A})$, and the ideal point corresponds to the assignment such that $c_A(x) = c_A^*, c_B(x) = c_B^*$.

The Kalai–Smorodinsky solution for such a bargaining problem is given by

$$x^{KS} = \arg\max\left\{f_A(x) : \frac{f_A(x)}{c_{A|B} - c_A^*} = \frac{f_B(x)}{c_{B|A} - c_B^*};\ x \in \mathcal{A}\right\}$$

Consider now LEAP and its optimal solution x^*. Due to Proposition 8, it follows that $x^* = \arg\min\{r_A(x) : r_A(x) = r_B(x) : x \in \mathcal{A}\}$. Then, by definition of f_A and f_B it turns out that:

$$\begin{aligned} x^* &= \arg\min\{r_A(x) : r_A(x) = r_B(x);\ x \in \mathcal{A}\} \\ &= \arg\max\{1 - r_A(x) : 1 - r_A(x) = 1 - r_B(x);\ x \in \mathcal{A}\} = x^{KS}. \end{aligned}$$

In conclusion the following property holds.

Observation 13: The optimal solution x^* of LEAP corresponds to a Kalai–Smorodinsky solution $\phi(F, d)$, where $F = \{(c_{A|B} - c_A(x), c_{B|A} - c_B(x)) : x \in \mathcal{A}\}$ and $d = (0, 0)$.

4.4 Bounds for the equilibrium problem

In this section we discuss procedures to obtain a good upper bound for EAP. Such bound is determined by simply rounding an optimal fractional solution x^* of LEAP which produces two different integer solutions (i.e., feasible assignments) x' and x''. We next describe such procedure and show some properties of the upper bound obtained. First, we need to provide a geometric characterisation of the two solutions obtained with the rounding procedure: to this purpose we make use of the graph and the matrix representations of feasible assignments. We then show some properties of the associated costs and relate them to the set of PO solutions.

Finally, in Section 4.4.3 we give the expression of the lower bound used in our enumeration scheme.

4.4.1 Rounding procedure

Any solution of an AP, i.e., an extreme point of $\mathcal{A}$, can be represented by a subset of pairwise disjoint arcs in a complete bipartite graph, which we refer to as G, with all the arcs ij, $i \in I_A \cup I_B$, $j \in M$, with cost c_{ij}. (We use the terms *arcs* and *nodes* with reference to an undirected graph, to distinguish them from vertices and edges of polyhedra.)

Similarly, recalling Proposition 12, an optimal solution of LEAP and, in general, any point lying on an edge of the polytope $\mathcal{A}$, may be represented as a 2-*matching*, that is, a subgraph of G where each node has degree at most two. We will prove that, in our case, such a 2-matching is the union of a standard matching and a single, possibly empty, cycle.

Call $G(x^*)$ the subgraph of G representing an optimal solution x^* of LEAP. $G(x^*)$ contains all the nodes of G (i.e., $I_A \cup I_B \cup M$) and the arcs ij such that $x^*_{ij} > 0$. The number of arcs of $G(x^*)$ incident to node $i \in I_A \cup I_B (j \in M)$ equals the number of fractional components of row i (or column j) in the matrix representation of x^*. As stated in Proposition 12, if a row i (or column j) has non integer components, then there are exactly two of such components whose values are equal to, say, f and $1-f$, with $0 < f < 1$. As a consequence, every node $i \in I_A \cup I_B$ (or $j \in M$) in $G(x^*)$ has degree at most two.

Let us denote each arc in $G(x^*)$ corresponding to a variable having value $0 < f \leq 1$ as a *f-arc*.

Consider the subgraph of $G(x^*)$ induced by all the nodes having degree two. It is not hard to see that this subgraph must be a collection of even cycles with alternating f and $(1-f)$- arcs. In conclusion, $G(x^*)$ is given by a collection of pairwise disjoint arcs with value 1 (1-arcs) and *alternating cycles* with f-arcs and $(1-f)$-arcs (corresponding to the fractional variables).

In the following proposition we will prove a property of $G(x^*)$ useful for constructing the rounded solutions.

Proposition 14: *The graph $G(x^*)$ representing an optimal solution x^* of LEAP contains at most one cycle.*

Proof: In the trivial case when x^* is integer, the set of edges of $G(x^*)$, denoted as $E(G(x^*))$, is a matching and there are no cycles.

Suppose x^* has non-integer components and, by contradiction, that $G(x^*)$ contains at least two cycles. Let $C_1 \subset E(G(x^*))$ be one of such cycles, and $C_2 \subset E(G(x^*))$ be the set of the edges of all the other possible cycles ($C_2 \neq \emptyset$). The components of x^* corresponding to the edges of $C_1 \cup C_2$ are fractional while all the other components are integer.

Given a cycle $C \subseteq E(G(x^*))$, since arcs in C have alternating values f and $1-f$, let us call F_C the set of f-arcs of C and $\overline{F}_C$ the set of $(1-f)$-arcs in C. Additionally, let H be the set of 1-arcs.

Assume first the case when all the f values are the same in the different cycles. Then x^* is as follows:

$$x^*_{ij} = \begin{cases} f & \text{if } ij \in F_{C_1} \cup F_{C_2}, \\ (1-f) & \text{if } ij \in \overline{F}_{C_1} \cup \overline{F}_{C_2}, \\ 1 & \text{if } ij \in H, \\ 0 & \text{otherwise.} \end{cases}$$

Consider now the four vertices x', x'', x^L, and x^R of $\mathcal{A}$, defined as follows:

$$x^*_{ij} = \begin{cases} 1 & \text{if } ij \in H \cup F_{C_1} \cup F_{C_2}, \\ 0 & \text{otherwise.} \end{cases} \quad x''_{ij} = \begin{cases} 1 & \text{if } ij \in H \cup \overline{F}_{C_1} \cup \overline{F}_{C_2}, \\ 0 & \text{otherwise.} \end{cases}$$

$$x^L_{ij} = \begin{cases} 1 & \text{if } ij \in H \cup \overline{F}_{C_1} \cup F_{C_2}, \\ 0 & \text{otherwise.} \end{cases} \quad x^R_{ij} = \begin{cases} 1 & \text{if } ij \in H \cup F_{C_1} \cup \overline{F}_{C_2}, \\ 0 & \text{otherwise.} \end{cases}$$

We have $x^* = fx' + (1-f)x''$ and x', x'' are the two adjacent vertices of $\mathcal{A}$ that define the edge where x^* lies. Consider now the point $x^M = \frac{1}{2}x' + \frac{1}{2}x''$. Note that x^M lies on the same edge where x^* lies. But x^M can also be expressed as $x^M = \frac{1}{2}x^L + \frac{1}{2}x^R$, i.e., this point is a convex combination of both x', x'' and x^L, x^R. But x^L and x^R are feasible assignments, and thus vertices of $\mathcal{A}$. Consequently, x^M and therefore x^*, do not lie on an edge of $\mathcal{A}$ which contradicts Proposition 12. Hence C_2 must be empty.

Similar arguments hold for the case where the f values are different for distinct cycles. □

On this grounds, the component ij of an optimal solution x^* of LEAP is equal to:

$$x^*_{ij} = \begin{cases} f & \text{if } ij \in F_{C^*}, \\ (1-f) & \text{if } ij \in \overline{F}_{C^*}, \\ 1 & \text{if } ij \in H, \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

where C^* is the single, possibly empty, cycle defined by the fractional components in x^*. When x^* is not integer, we may obtain two different feasible matchings, x' and x'', with the following rounding procedure,

$$x'_{ij} = \begin{cases} 1 & \text{if } ij \in H \cup F_{C^*}, \\ 0 & \text{otherwise.} \end{cases} \quad x''_{ij} = \begin{cases} 1 & \text{if } ij \in H \cup \overline{F}_{C^*}, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

It is easy to see that one of these assignments is feasible for Problem 8 while the other is not (i.e., the constraint $r_A(x) \geq r_B(x)$ is violated).

4.4.2 Quality of the rounded solutions

To illustrate the characteristics of the solutions obtained using the rounding procedure we need to refer to their mapping into the (c_A, c_B) space. When x^* is not integer, then

$x' \neq x^* \neq x''$. In this case, we assume, with no loss of generality, that $r_A(x') < r_B(x')$, and $r_A(x'') > r_B(x'')$.

Observation 15: The mapping of an optimal solution x^* of LEAP on the (c_A, c_B)-plane lies on the segment whose extreme points are $(c_A(x'), c_B(x'))$ and $(c_A(x''), c_B(x''))$.

Proposition 16: *The rounded solutions x' and x'' are Pareto-optimal.*

Proof: We prove that x' is Pareto-optimal. The result holds, by symmetry, for x''.

Let us suppose, by contradiction, that x', for which $r_A(x') < r_B(x')$, is dominated. Then there exists at least one feasible assignment $\hat{x}$ that dominates x', and we have:

$$c_A(\hat{x}) \leq c_A(x'), c_B(\hat{x}) < c_B(x') \quad \text{or} \tag{12}$$

$$c_B(\hat{x}) \leq c_B(x'), c_A(\hat{x}) < c_A(x'). \tag{13}$$

Suppose, without loss of generality, that equation (12) holds. Given Propositions 8 and 9, we have $c_B^* < c_B(\hat{x}) < c_B(x')$ and $r_A(\hat{x}) < r_B(\hat{x})$. Consider now x'', for which $r_A(x'') > r_B(x'')$, and the solution $\tilde{x} = \lambda\hat{x} + (1-\lambda)x''$, with $0 < \lambda < 1$. Therefore, $c_A(\tilde{x}) = \lambda c_A(\hat{x}) + (1-\lambda)c_A(x'')$ and $c_B(\tilde{x}) = \lambda c_B(\hat{x}) + (1-\lambda)c_B(x'')$. Hence, due to equation (12), $c_B(\tilde{x}) < \lambda c_B(x') + (1-\lambda)c_B(x'') = c_B(x^*)$. From definition of $r_B(x)$, this implies $r_B(\tilde{x}) < r_B(x^*)$; and since, from Proposition 8, $r_A(\tilde{x}) = r_B(\tilde{x})$ and $r_A(x^*) = r_B(x^*)$, it follows that $r_A(\tilde{x}) < r_A(x^*)$ which contradicts that x^* is optimal for LEAP. □

Proposition 17: *The two solutions provided by the rounding procedure are efficient points.*

Proof: We have to prove that, in the (c_A, c_B)-space, no PO points lie below the line passing through the projections of x' and x'' (as defined above) in the cost space. By contradiction, suppose that there is such a point y: then either $r_A(y) < r_B(y)$ or $r_A(y) > r_B(y)$, since it cannot be $r_A(y) = r_B(y)$, otherwise x^* would not be optimal for LEAP. Suppose that $r_A(y) > r_B(y)$, we may thus identify a point $\tilde{x} = \lambda y + (1-\lambda)x'$ such that $\tilde{x} \in \mathcal{A}$ and $r_A(\tilde{x}) = r_B(\tilde{x})$, for some $0 < \lambda < 1$. Then $\tilde{x}$ would also satisfy $r_A(\tilde{x}) < r_A(x^*)$, contradicting that x^* is an optimal solution of LEAP. The case $r_A(y) < r_B(y)$ is symmetrical. □

As a consequence of this result, given an optimal fractional solution x^* of LEAP, x'' is an upper bound to problem EAP and therefore the only possible optima of EAP are the assignments x such that $c_A(x^*) \leq c_A(x) \leq c_A(x'')$ and $c_B(x'') \leq c_B(x) \leq c_B(x^*)$, and are above the segment connecting the projections of x^* and x'' in the cost space.

4.4.3 Lower bound for the equilibrium problem

As mentioned in Section 1, research on constrained APs (Aggarwal, 1985; Brans et al., 1972; Kennington and Mohammadi, 1994; Mazzola and Neebe, 1986; Rosenwein, 1991) often makes use of Lagrangean relaxations to produce strong lower bounds.

The EAP is a standard AP with the single additional constraint $r_A(x) \geq r_B(x)$. When such a constraint is dualised in a Lagrangean relaxation, the problem satisfies the *Geoffrion's integrality property* (Geoffrion, 1974) since the feasible region is the assignment polytope. As a consequence, the best Lagrangean bound cannot be better than that obtained through linear relaxation. Therefore, in the branch-and-bound scheme to solve EAP, we make use of the lower bounds provided by the LP's.

In the case of integer costs c_{ij}^A and fractional solution x^* of LEAP, we may further improve these bounds using the following lower bound:

$$r_A^* \geq \frac{\lceil c_A(x^*) \rceil - c_A^*}{c_{A|B} - c_A^*}.$$

5 Computational experience

Our branch-and-bound scheme (in the following referred to as EAP_bb) has been implemented using the bounds described in Sections 4.4 and 4.4.3 in the default CPLEX 8.0 branching scheme via callback functions (see ILOG Inc., 2003). We compared the solution produced by our algorithm with those by the default CPLEX branch and cut on a series of randomly generated test instances, varying the number of jobs for each agent (n_A, n_B) and the intervals from which costs are generated.

In a first set of experiments, whose results are reported in Table 1, we compared our algorithm and the default CPLEX branch and cut implementation, in terms of CPU times and effectiveness. Each one of the eight classes has the same size (which is reported in the first column). The results are average values over 30 instances for each class (three disjoint cost ranges were considered and, for each cost range, ten different randomly generated instances were solved). The average CPU times (in seconds) needed by EAP_bb and default CPLEX 8.0 are reported. Results marked with '*' indicate that the average values are only relative to those subsets of instances where the optimal solution is obtained within a given time limit of 3600 s; whereas mark '**' indicates that the algorithm did not reach the optimum within the same time limit, and the average times needed to find the best feasible solutions are presented.

Table 1 Time comparisons between EAP_bb and CPLEX 8.0. The results are relative to averages over different cost ranges and randomly generated instances

Problem	*Time* (s)		*Time*
size	*EAP_bb*	*CPLEX*	*improvement* (%)
40, 40	1.36	130.39	0.99
50, 50	2.2	85.81	0.97
60, 60	3.66	170.91*	0.98
80, 80	5.09	534.74*	0.99
100, 100	50.31	534.49*	0.91
200, 200	97.04	1865.95*	0.95
250, 250	101.2	**	>0.98
300, 300	150.0	**	>0.98

EAP_bb improves the computational times upon CPLEX significantly: this is particularly evident for the largest instances of problems; our algorithm takes, on the average, less than 3 min to obtain an optimal solution while CPLEX takes hours.

In Table 2, the number of nodes of the branch-and-bound tree (measuring memory occupation) and CPU times are reported. Moreover the results relative to different cost ranges are presented on separate rows. We may notice how using improved upper and lower bounds turns out in a dramatic decreasing of the number of nodes generated in the enumeration scheme. More surprisingly, in both algorithms, the number of nodes is highly sensitive to the cost ranges.

Again, also in Table 2, mark '*' indicates that we report average values only relative to the instances where the optimal solution is obtained within a given time limit of 3600 s; whereas mark '**' indicates that the algorithm did not reach the optimum within the same time limit and the average times needed to find the best feasible solutions are presented.

Table 2 Time and memory comparisons between EAP_bb and CPLEX 8.0. The results are relative to averages over randomly generated instances

Problem	*Costs*	*Nodes of B&B tree*		*CPU time*	
size	*range*	*EAP_bb*	*CPLEX*	*EAP_bb*	*CPLEX*
40, 40	1–50	1	114012	0.58	384.62
	1–100	6	240	0.82	2.48
	1–200	36	188	2.69	4.07
50, 50	1–50	4	212620	2.15	230.05
	1–100	24	480	2.61	6.35
	1–200	44	500	3.23	6.76
60, 60	1–50	16	153360*	2.78	383.9*
	1–100	6	15180*	2.72	118.01*
	1–200	65	720*	5.47	10.88*
80, 80	1–50	1	83450*	3.79	1204.71*
	1–100	3	14450*	6.38	237.11*
	1–200	13	10840*	5.11	162.4*
100, 100	1–50	46.5	2*	19.19	131.65*
	1–100	1	505*	8.36	26.76*
	1–200	13	904*	24.52	904.07*
200, 200	1–100	2	18000*	93.12	2321.88*
	1–200	18	9533*	100.96	1410.03*

6 Testing the equilibrium strategy

In this section we analyse the quality of the equilibrium solution for CAP. Rather than addressing the efficiency of the algorithm used to find the equilibrium solution, we try to evaluate the suitability of the equilibrium approach in a 2-agents competition. In other words, one may question why an agent should find convenient to settle on an equilibrium solution. Here we try to set a specific experimental framework and draw some partial conclusions from the results.

We assume that two agents compete for the assignment of resources in a dynamic environment, where they are given more that one instance of the AP, and that the costs related with this assignment vary for each instance. We also refer to each instance of the problem as a *game*. Such situation may well represent realistic settings where the costs of consecutive games (biddings, market shares, task assignments) are affected from external factors that are out of the agents' control. In these settings the agents have to choose a *strategy*, that is, a rule to solve each instance of the assignment.

In our experiments we compared the following three simple strategies:

- *Equilibrium Strategy* (ES), where the two agents agree over the equilibrium solution, that is, an EAP is solved for each game.
- *Global Optimal Strategy* (OS), where the agents agree to choose the assignment with global minimum cost, eventually sharing the differential benefits at the end of each game.
- *Alternate Strategy* (AS), where the two agents solve the AP sequentially, the first one choosing the resources in an optimal way among all available resources, and, the second among those that are not used by the first one. In this case, the first decisor is selected by tossing a coin before each game.

We have set the following experiments. First, we generate a number of games where the assignment costs for the two agents are chosen at random according to the uniform distribution in a given interval. Second, we solve each game using the three strategies. Third, we compare the costs for the two agents (A and B) considering their average and variance over all the generated games.

We have considered different sizes of the APs, expressed as the number of resources to be assigned to the two agents, namely, 50, 100, 150, and 200 (in the following referred to as n); for each possible size the assignment costs have been generated randomly in the intervals $(0, 4n)$ ('small') and $(0, 8n)$ ('large'). For each size and cost range, we have generated 300 APs, and solved them with the three strategies. The results are summarised in Table 3, where the first column reports the size of the AP (n) and the second column the interval used to generate the costs. The following three columns report the average cost for Agent A under the ES, OS and AS strategies. Columns six to eight report the same data for Agent B.

Obviously, as it might be expected, on the average the OS is the best in terms of final costs. It should be noticed that the average values of the costs for the two agents are also very close, under this strategy. The reason of this fact is related to the high number of instances (300) over which average are computed and to the uniform distribution used for the assignment costs c_{ij}. In a real setting, when the number of games between two competitors is limited, this would be equivalent to require to transfer some of the utility at given time intervals, i.e., it assumes that the utility of the games is fully represented by the costs of the assignment. This is in fact a strong assumption in most cases.

The results highlight that, on the average, the distance between the Global Optimal and the Equilibrium strategies is actually small. An interpretation of this fact leads to conclude that the ES, in the long run, is not significantly worse than the OS in terms of global costs; but it is, by far, easier to implement in a competitive setting.

The AS, on the contrary, exhibits the largest costs for all problem sizes and cost types. Thus, in the long run and if agents have equal probability to win, it is convenient to settle on the equilibrium solution rather than simply compete to be first in the decision making process. Additionally, a strategy that exhibits low variance may be preferable to another that does not: as, in most cases, decision makers are interested in reducing uncertainty and having a steady behaviour in different games.

In Table 4, the information about the cost standard deviations, for the same set of experiments of Table 3, are reported. Not surprisingly, the ES performs with the lowest standard deviation, and results are more stable even when compared to the optimal one. It is important to stress that these comparisons are related with the average costs of A and B over a significantly large number of games (300). However, with reference to the results reported here, one may conclude that an ES, at least in the long run, stands as an interesting, perhaps easier to implement in a real world setting, alternative position with respect to total competition (AS) or total collaboration (the OS) scenarios.

Table 3 Average costs for Agents A and B under Equilibrium Strategy (ES), Global Optimal Strategy (OS), and Alternate Strategy (AS) Strategies

		Agent A			*Agent B*		
n	*Cost range*	*ES*	*OS*	*AS*	*ES*	*OS*	*AS*
50	Small	175.33	170.40	218.23	172.50	171.92	230.99
50	Large	337.26	332.91	415.62	329.56	322.32	436.60
100	Small	350.31	349.27	464.11	346.90	342.94	447.75
100	Large	679.06	670.49	905.48	671.67	668.12	871.18
150	Small	527.35	525.67	718.88	527.16	523.54	663.40
150	Large	1010.38	1003.61	1391.70	1022.33	1018.34	1295.90
200	Small	705.54	705.57	941.70	708.80	703.73	925.32
200	Large	1363.24	1358.20	1819.37	1354.98	1349.13	1776.77

Table 4 Cost standard deviations for Agents A and B under Equilibrium Strategy (ES), Global Optimal Strategy (OS), and Alternate Strategy (AS) Strategies

		Agent A			*Agent B*		
n	*Cost range*	*ES*	*OS*	*AS*	*ES*	*OS*	*AS*
50	Small	24.47	26.33	103.73	25.10	28.15	103.43
50	Large	49.44	52.76	199.84	45.86	52.23	198.49
100	Small	33.38	37.71	210.63	33.17	34.03	205.68
100	Large	72.68	73.51	428.95	74.44	82.81	414.93
150	Small	42.99	48.91	320.52	42.36	45.77	311.87
150	Large	92.16	97.81	636.13	76.75	88.59	626.76
200	Small	46.11	54.15	418.84	47.84	52.05	427.11
200	Large	96.34	110.83	825.48	89.70	108.59	851.67

7 Conclusions and future works

In this paper we introduce and analyse a variation of the AP that arises in the context of Multi-Agent Systems. We address the case of two agents competing for the usage of a common set of resources, where each agent has its own objective of minimising the total assignment cost of its jobs only.

We define and analyse the problem of determining Pareto-optimal solutions, in order to provide a basis over which the agents can negotiate, and the problem of finding an *Equilibrium Assignment*, i.e., a specific solution that represents for both the agents the 'best compromised assignment' providing some sort of equity in the deviations from the optimal *unconditioned* solutions, that would happen if each agent were assigned its optimal set of resources. We show that this solution is basically equivalent to one proposed by Kalai and Smorodinsky for the bargaining problem in game theory. The EAP is formulated as an integer linear problem. By a suitable linear relaxation, we develop good lower and upper bounds for EAP. An efficient and effective branch-and-bound algorithm that exploits these bounds is also proposed.

This work puts several directions forward for future research. One is the design of a proper heuristic to efficiently enumerate all Pareto-optimal solutions. Another is to extend EAP to other types of equilibria, such as those arising when agents are given different 'weights' based on their 'power' in the bargaining game. Finally, different scenarios may be considered, including cooperation between agents where benefits of cooperation are distributed from one agent to the other.

Acknowledgements

The authors wish to gratefully acknowledge the helpful comments and constructive suggestions of the editor and two anonymous referees.

References

Aboudi, R. and Nemhauser, G. (1991) 'Some facets for an assignment problem with side constraints', *Operations Research*, Vol. 39, pp.244–250.

Aggarwal, V. (1985) 'A Lagrangean-relaxation method for the constrained assignment problem', *Computers and Operations Research*, Vol. 12, pp.97–106.

Agnetis, A., Mirchandani, P.B., Pacciarelli, D. and Pacifici, A. (2004) 'Scheduling problems with two competing agents', *Operations Research*, Vol. 52, No. 2, pp.229–242.

Arbib, C. and Rossi, F. (2000) 'Optimal resource assignment throught negotiations in a multi-agent manufactoring system', *IIE: Transaction*, Vol. 32, No. 32, pp.963–974.

Benders, J.F. and van Nunen, J.A.E.E. (1983) 'A property of assignment type mixed integer linear programming problems', *Operations Research Letters*, Vol. 2, No. 2, pp.47–52.

Bertsekas, D. (1981) 'A new algorithm for the assignment problem', *Mathematical Programming*, Vol. 21, pp.152–171.

Brans, J., Leclerq, M. and Hansen, P. (1972) *An Algorithm for Optimal Reloading of Pressurized Water Reactors*, North-Holland Publishing, M.Ross edition, Amsterdam.

Burkard, R.E. (2002) 'Selected topics in assignment problems', *Discrete Applied Mathematics*, Vol. 123, pp.257–302.

Burkard, R.E. and Cela, E. (1999) 'Linear assignment problems and extensions', in Du, D-Z. and Pardalos, P.M. (Eds.): *Handbook of Combinatorial Optimization*, Kluwer Academic Publishers, Suppl. Vol. A, pp.75–149.

Cattryse, D.G. and Wassenhove, L.N. (1992) 'A survey of algorithms for the generalized assignment problem', *European Journal of Operation Research*, Vol. 60, No. 3, pp.260–272.

Conley, J.P. and Wilkie, S. (1991) 'The bargaining problem without convexity: extending the egalitarian and Kalai–Smorodinsky solutions', *Economics Letters*, Vol. 36, No. 4, pp.365–369.

Conley, J.P. and Wilkie, S. (1996) 'An extension of the Nash bargaining solution to nonconvex problems', *Games and Economic Behavior*, Vol. 13, No. 1, pp.26–38.

Felici, G., Mecoli, M., Mirchandani, P.B. and Pacifici, A. (2004) *Assignment Problem with Two Competing Agents*, Technical Report, Istituto di Analisi dei Sistemi ed Informatica del CNR, Viale Manzoni 30, Roma, Italy.

Fisher, M.L., Jaikumar, R. and Wassenhove, L.N. (1986) 'A multiplier adjustment method for the generalized assignment problem', *Management Science*, Vol. 32, No. 9, pp.1095–1103.

Gabow, H.N. and Tarjan, R.E. (1989) 'Faster scaling algorithms for network problems', *SIAM Journal on Computing*, Vol. 18, No. 5, pp.1013–1036.

Garey, M.R. and Johnson, D.S. (1979) *Computers and Intractability. A Guide to the Theory of NP-Completness*, WH Freman, New York.

Geoffrion, A. (1974) 'Lagrangean relaxation for integer programming', *Mathematical Programming Study*, Vol. 2, pp.82–114.

Gupta, A. and Sharma, J. (1981) 'Tree search method for optimal core mamangement of pressurized water reactors', *Computers and Operations Research*, Vol. 8, pp.263–266.

Herrero, M.J. (1989) 'The Nash program: non-convex bargaining problems', *Journal of Economic Theory*, Vol. 49, No. 2, pp.266–277.

Higgins, A.J. (2001) 'A dynamic tabu search for large-scale generalized assignment problems', *Computers and Operations Research*, Vol. 28, No. 10, pp.1039–1048.

Hougaard, J.L. and Tvede, M. (2003) 'Nonconvex n-person bargaining: efficient maxmin solutions', *Journal of Economic Theory*, Vol. 21, No. 1, pp.81–95.

ILOG Inc. (2003) *ILOG CPLEX 8.0 User Manual*. Gentilly Cedex, France.

Kalai, E. and Smorodinsky, M. (1975) 'Others solutions to Nash's bargaining problem', *Econometrica*, Vol. 43, pp.513–518.

Kaneko, M. (1980) 'An extension of the Nash bargaining problem and the Nash social welfare function', *Theory and Decision*, Vol. 12, pp.135–148.

Kennington, J.L. and Mohammadi, F. (1994) 'The singly constrained assignment problem: a lagrangian relaxation heuristic algorithm', *Computational Optimization and Applications*, Vol. 3, pp.7–26.

Khun, A.W. (1955) 'The Hungarian method for the assignment problem', *Naval Research Logistic Quart.*, Vol. 2, pp.83–97.

Lin, Z. (1997) 'The Nash bargaining theory with non-convex problems', *Econometrica*, Vol. 65, No. 3, pp.681–686.

Mariotti, M. (1998) 'Nash bargaining theory when the number of alternatives can be finite', *Social Choice and Welfare*, Vol. 15, Springer, Berlin, Heidelberg.

Mazzola, R.A. and Neebe, A.W. (1986) 'Resource constrained assignment scheduling', *Operations Research*, Vol. 34, No. 4, pp.560–572.

Nash, J.F. (1950) 'The bargaining problem', *Econometrica*, Vol. 18, pp.155–162.

Nemhauser, G. and Wolsey, L. (1988) *Integer and Combinatorial Optimization*, Wiley-Interscience Series in Discrete Mathematics and Optimization, Wiley, NEM g 88:1 P-Ex.

Oliveira, E., Fischer, K. and Stepankova, O. (1999) 'Multi-agent systems: which research for which applications', *Robotics and Autonomous Systems*, Vol. 27, pp.91–106.

Punnen, A.P. and Aneja, Y.P. (1995) 'A tabu search for the resource-constrained assignment problem', *Journal of Operational Research*, Vol. 46, pp.214–220.

Rosenwein, M.B. (1991) 'An improved bounding procedure for the constrained assignment problem', *Computers and Operations Research*, Vol. 18, No. 6, pp.531–535.

Ross, G.T. and Soland, R.M. (1975) 'A branch and bound algorithm for the generalized assignment problem', *Mathematical Programming*, Vol. 8, pp.91–103.

Savelsbergh, M. (1997) 'A branch-and-price algorithm for the generalized assignment problem', *Operations Research*, Vol. 45, No. 6, pp.831–841.