On scheduling fees to prevent merging, splitting and transferring of jobs

Herve Moulin *Rice University*

May 2004

Abstract

A deterministic server is shared by users with identical linear waiting costs, requesting jobs of arbitrary lengths. Shortest jobs are served first for efficiency. The server can monitor the length of a job, but not the identity of its user, thus merging, splitting or partially transferring jobs offer cooperative strategic opportunities. Can we design cash transfers to neutralize such manipulations?

We prove that merge-proofness and split-proofness are not compatible, and that it is similarly impossible to prevent all transfers of jobs involving three agents or more. On the other hand, robustness against pairwise transfers is feasible, and essentially characterize a one-dimensional set of scheduling methods. This line is borne by two outstanding methods, the merge-proof S^+ and the split-proof S^-.

Splitproofness, unlike Mergeproofness, is not compatible with several simple tests of equity. Thus the two properties are far from equally demanding.

Key words: scheduling, queuing, merging, splitting, transferring, linear waiting cost.

Acknowledgements: I am grateful for their hospitality during the preparation of this paper to the University of Auckland, and the Facultes Universitaires Notre Dame de la Paix in Namur, in particular to Arkadii Slinko and Francois Maniquet.

This work is supported by the NSF under grant SES-0414543.

1 The problem and the punch lines

Dividing the burden of joint externalities raises many issues of incentive-compatibility. One of these is strategic transferring, merging, or splitting of certain private characteristics of the participants. This type of manipulation is discussed in the fair division literature (see details in section 2); here we study it in a simple scheduling problem with transferable utility.

A single deterministic server/machine is shared by users with linear waiting costs, requesting jobs of arbitrary lengths. A job of size x_i takes x_i units of time to process; an agent's disutility is the waiting time until her job is completed, augmented by a (positive or negative) cash payment selected by the mechanism. The key assumptions is that the server can monitor the size of a job, but not the identity of its user. This creates opportunities for manipulation if two agents i, j can costlessly merge two jobs of sizes x_i, x_j into a single job of size $x_i + x_j$, reporting to the server under one of their names; or if agent i can split his job of size x_i into two smaller jobs x_i^1, x_i^2 with $x_i^1 + x_i^2 = x_i$, then request service under two aliases; or, finally, if agent i can transfer a fraction of his job and add it to agent j's job.

The key assumption is realistic when the usage of the server/machine is private, and can't be traced to its actual beneficiary. Think of a tool that agents carry to their private workstation, for instance a software used on a private machine. Protecting the privacy of the users is often a design constraint, e.g., when they share a single "link" (access point to a database or phone line). Then the needs of each user of the link remain unknown to the server, who cannot detect if and when the link is used by agent i on behalf of another agent j. Two more factors affect the feasibility of the strategic maneuvers in question. First assuming a false identity should be easy, as is the case in huge networks such as the internet, where protecting the system performance against aliases is an important design issue: Douceur [2002]. The second factor is the cost of merging, splitting or transferring jobs: it is minimal if the job produces an electronic document, or a physical tool easily transported from one job to the next.

Two very simple scheduling mechanisms illustrate the cooperative manipulations that we wish to prevent. Given identical linear waiting costs and the feasibility of cash transfers, efficiency requires to serve the shortest jobs first. Suppose the server does this and performs no monetary transfer (at least when the efficient scheduling order is unique, i.e., all jobs are of different size). This mechanism is highly vulnerable to splitting: given two real jobs $x_1 = 4$, $x_2 = 3$, agent 1 splits his job as $x'_1 = x''_1 = 2$ and cuts his waiting time by 3. Partial transfers may also work: say we have three jobs $(x_1, x_2, x_3) = (1, 4, 5)$; if agent 3 transfers 2 units of her job to agent 1, resulting in $(x'_1, x_2, x'_3) = (3, 4, 3)$, she will complete x_3 before agent 2 is served, and the net gain \$4 can be divided between agents 1 and 3 (we assume a cost of \$1 per unit of time). But the merging of jobs is clearly not profitable, as this can only delay their completion.

Consider next a mechanism serving the *longest* jobs first, thus *maximizing* total waiting cost. No matter how it deals with ties, this mechanism is badly vulnerable to merging, as well as to partial transfers: simply use the above examples backward. But the splitting of a job is never profitable.

Can we design a system of cash transfers to prevent in all problems single agents from splitting their job, and coalitions from merging them under a single identity? And what about partial transfers of jobs?

Despite the simplicity of our scheduling model, some of the answers to these questions are disappointingly negative. If the potential set of users contains at least 4 agents, a mechanism treating equals equally cannot be both *merge-proof and split-proof:* Theorem 1 in Section 4. Moreover *every* continuous mechanism (i.e., net waiting costs depend continuously upon the profile of job sizes) is vulnerable to transfers involving three agents or more: Section 8.

Yet we show here that the family of merge-proof scheduling mechanisms is fairly large, and so is that of split-proof mechanisms. Moreover, each family contains many mechanisms immune to job transfers involving only two agents. Another result is that immunity to splitting is much more demanding than to merging. Specifically we show that Split-proofness, unlike Merge-proofness, is incompatible with several compelling fairness requirements. Proposition 1 in Section 5 gives a precise content to this statement. Restrict attention to efficient mechanisms (serving successively jobs of increasing size) treating equals equally, and continuous. Every splitproof mechanism must then charge a positive fee to null jobs, who create no externality whatsoever; it must also *subsidize* some jobs in the sense that their net waiting cost is smaller than their size x_i; next,the net cost of agent i is not always weakly increasing in x_i; the ordering of net costs must sometime contradict that of job lengths; and finally the net waiting cost of a given job is unbounded when other jobs become arbitrarily large. By contrast, merge-proofness is compatible with all five properties just described.

In Section 6, we construct a large family of efficient scheduling mechanisms, treating equals equally and continuous, and for which the role of merge-proofness and split-proofness is especially easy to describe. Pick a continuous function θ from $\mathbb{R}^2_+$ into $\mathbb{R}$ such that $\theta(a,b)+\theta(b,a) = \min\{a,b\}$ for all a,b. Label the set of users $N = \{1,2,..,n\}$ in such a way that $x_1 \leq x_2 \leq .. \leq x_n$. The θ-mechanism serves the job in the efficient order $1,2,..,n$, and performs cash transfers resulting in the net waiting cost $y_i = x_i + \sum_{j \neq i} \theta(x_i, x_j)$ for all i. By construction of θ, this implies $\sum_i y_i = nx_1 + (n-1)x_2 + ... + x_n$, so these transfers are balanced.

We call the above mechanism *separable* because it divides the efficient externality $\min\{x_i, x_j\}$ between agents i,j without paying attention to other job lengths. Proposition 2 in Section 6 characterizes merge-proof separable methods by a system of inequalities slightly less demanding than the super-additivity of θ in its first variable, and split-proof separable methods by a similar system slightly more demanding than the sub-additivity of θ in its first variable.

Two separable mechanisms stand out. The first one, called S^+, splits the (i,j)-externality equally, namely $\theta^+(a,b) = \frac{1}{2}\min\{a,b\}$. The second mechanism, called S^-, uses the function $\theta^-(a,b) = b - \frac{1}{2}\max\{a,b\}$. The method S^+ corresponds to the Shapley value of the *optimistic* stand alone cooperative game (a coalition S standing alone is served before N/S); the method S^- to the Shapley value of the *pessimistic* stand alone cooperative game (a coalition S standing alone is served after N/S).

We find that S^+ is merge-proof, whereas S^- is split proof - hence the latter shares all unpalatable consequences of splitproofness discussed above.

In Section 7 we turn to the strategic transfer of jobs. We restrict attention to job transfers involving only two agents, combined with cash transfers within a coalition of arbitrary size. We show that S^+, S^- as well as their affine combinations $y = a \cdot y^+ + (1-a) \cdot y^-, a \in \mathbb{R}$, are immune to such manipulations. Our main result, Theorem 2, is a characterization of the line of methods borne by S^+ and S^- based on this property of *pairwise transfer-proofness*. Then we characterize the S^+ method either by the requirement that null jobs should not pay (or receive) anything, or by ruling out subsidies beyond the optimistic stand alone wait ($x_i \leq y_i$).

2 Related literature

The earliest discussion in the fair division literature of manipulation by merging, splitting, and transferring, is in the rationing problem: each agent has a claim/liability over an amount of money smaller than the sum of individual claims/liabilities. If the claims take the form of anonymous, transferable bonds, dividing the money in proportion to individual claims is the only method invulnerable to transfers, as well as to merging or splitting: Banker [1981]. Variants and extensions of this result are in Moulin [1987], DeFrutos [1999], and Ju [2003]. Related properties of transfer-proofness appear in the quasi-linear social choice problem (Moulin [1985], Ermolov [1995], Chun [2000]), in axiomatic cost-sharing (Sprumont [2004]) and more: Ju and Miyagawa [2003] offer a unified treatment of most of this literature.

We now review the recent and growing microeconomic literature on scheduling. A familiar extension of our model allows linear waiting costs to vary accross participants. A scheduling problem consists of a profile of job sizes x_i and waiting costs δ_i per unit of time. Agent $i's$ disutility is $\delta_i w_i + t_i$, where w_i is waiting time until completion of job i and t_i is the cash payment. Minimizing total waiting cost requires to serve the jobs in the increasing order of the ratios $\frac{x_i}{\delta_i}$ (Smith [1956]).

The mechanism designer can use the cash transfers to ensure truthful (dominant strategy) elicitation of the privately known waiting costs: utilities are linear in money (and waiting costs), therefore Vickrey-Clarke-Groves mechanisms can be readily applied. The first authors to explore this idea are Dolan [1978] and Mendelson and Whang [1990]. In fact, given linear waiting costs, we can construct a budget-balanced (fully efficient) VCG mechanism: Suijs [1996], Mitra and Sen [1998], Mitra [2001,2002]. If we must elicit job lengths instead of waiting costs, a similar construction is possible (Hain and Mitra [2001], Kittsteiner and Moldovanu [2003a,b]), provided the VCG mechanisms are suitably generalized to take into account the more complicated allocative externalities from misreporting the size of one's job.

In the linear scheduling model cash transfers are also a simple tool to achieve fairness, namely an equitable sharing of the congestion externality. Several auhors simply apply off-the-shelve solution concepts like the Shapley value or the core to a relevant cooperative game: Curiel et al. [1989], [1993], [2002], Hamers et al. [1996]. The most popular solution is the Shapley value of the *optimistic* stand alone cooperative game: Curiel et al. [1993] and Klijn and Sanchez [2002]. It plays an important role in the current paper as solution S^+. In the case of identical job sizes, this solution is axiomatized by Maniquet [2003], while Katta and Sethuraman [2004] suggest alternative interpretations of fairness. The Shapley value of the *pessimistic* stand alone game corresponds to our second solution S^-. With identical job sizes it is axiomatized by Chun [2004a]. Chun [2004b] allows for variable job sizes, extends both solutions to this context and offers parallel characterizations.

Our approach is original on two accounts. First we explore a new kind of cooperative manipulation, quite different from the misreport of waiting costs or of job sizes. In our model, individual preferences are known to the server, and job size is observable. All the action comes from the inability of the server to detect the true identity of users, and the users' ability to request a job, or part of a job, without revealing its true beneficiary.

Secondly we explore the compatibility of our strategy-proofness properties with four classic equity tests, based on monotonicity and bounds on individual disutilities (see Section 5). These tests are inspired by the fair division literature, and play a role as well in the work of Maniquet [2003] and Chun [2004]. Here they reveal a fundamental asymmetry between the requirements of merge-proofness and split-proofness (Proposition 1).

The companion paper Moulin [2004] discusses the same strategic maneuvers when the server instead of cash transfers, uses randomization. In that context, Split-proofness remains a much more demanding property than Merge-proofness, yet these two properties are now compatible. A certain probabilistic scheduling rule, the Proportional rule, is characterized by the combination of Merge-proofness, Split-proofness and a couple of natural properties of invariance and fairness.

3 The model

The set $\mathcal{N}$ contains all potential users of the simple machine. It may be finite or infinite. A scheduling problem involves a *finite* subset N of $\mathcal{N}$. Agent i's job is completed in exactly x_i units of machine-time. Given a scheduling problem (N, x), where $x \in \mathbb{R}_+^N$, the server must choose the ordering σ of N - the schedule - in which the jobs will be processed, and a vector $t \in \mathbb{R}^N$ of monetary transfers such that $\sum_N t_i = 0$.

Each agent incurs a waiting cost of \$1 per unit of time until completion of his/her job (a partially completed job is useless). The equality of waiting costs is an important simplifying assumption[1].

We write $\sigma(i) < \sigma(j)$ to mean that agent i precedes agent j in the ordering σ, and $P(i, \sigma) = \{j \in N / \sigma(j) < \sigma(i)\}$ is the set of agents preceding i in σ. Thus the disutility of agent i given σ and t is

$$y_i = x_i + \sum_{P(i,\sigma)} x_j + t_i \tag{1}$$

Notice that t_i is a tax on agent i when $t_i > 0$ and a subsidy when $t_i < 0$.

The standard notation $a_S = \sum_{i \in S} a_i$ will be used throughout the paper. Because monetary transfers are unrestricted, efficiency amounts to choose an ordering σ minimizing total waiting cost

$$\sum_N (x_i + \sum_{P(i,\sigma)} x_j) = x_N + \sum_{(i,j):\sigma(i)<\sigma(j)} x_i$$

An ordering is efficient if and only if it schedules shortest jobs first. The set of efficient orderings is

$$E(N, x) = \{\sigma | \text{for all } i, j \in N : x_i < x_j \Longrightarrow \sigma(i) < \sigma(j)\}$$

We use the notations $a \wedge b = \min\{a, b\}$, and $N(2)$ for the set of all non-ordered pairs $\{i, j\}$ of distinct agents. Then the minimal (efficient) total waiting cost $v(N, x)$ is

$$v(N, x) = x_N + \sum_{N(2)} x_i \wedge x_j$$

Definition 1 *Given $\mathcal{N}$, a scheduling **mechanism** μ associates to every problem (N, x), where $N \subset \mathcal{N}$ and $x \in \mathbb{R}_+^N$, a pair $\mu(N, x) = (\sigma, t)$, where σ is an ordering of N, and $t \in \mathbb{R}^N$ with $t_N = 0$. A scheduling **method** m associates to every problem (N, x) a profile of net waiting costs $m(N, x) = y, y \in \mathbb{R}^N$, such that*

$$y_N = x_N + \sum_{(i,j):\sigma(i)<\sigma(j)} x_i, \text{ for some ordering } \sigma \text{ of } N.$$

To each mechanism μ, we associate a method m by formula (1). We call the mechanism μ *efficient* if $\sigma \in E(N, x)$ for all N, x; we call the method m efficient if $y_N = v(N, x)$ for all N, x. To an efficient method m corresponds

[1] The consequences of relaxing this assumption are briefly discussed in Section 9.

essentially a unique efficient mechanism μ : the only qualification is at those problems x where some jobs have the same size, $x_i = x_j$, so that $E(N,x)$ is not a singleton. As this will cause no confusion, we shall state some of our axioms for mechanisms (e.g. Merge-proofness) and some of them for methods.

Our first normative requirement is the standard horizontal equity:

Equal Treatment of Equals (ETE): $x_i = x_j \implies y_i = y_j$, for all $(N,x), i,j \in N$

All methods discussed below meet ETE, yet this property is not necessary to our main characterization result (Theorem 2). By contrast, the following axiom plays a key role in Theorem 2

Continuity (CONT): the mapping $x \to y(N,x)$ is continuous on $\mathbb{R}_+^N$ for all N.

Continuity ensures that microscopic variations in the job sizes do not have a macroscopic impact on the profile of net waiting costs. In particular when $x_i = x_j$, a small tremble of x_i - the result of a measurement error, or of a strategic move - is not a matter of concern to agents i,j, or to anyone else.

Our first example is a natural discontinuous mechanism.

Example 1 Shortest Jobs First

For every (N,x) where $x_i \neq x_j$ for all i,j, the mechanism selects the unique efficient ordering σ and performs no transfers. At other profiles, it performs the minimal transfers required by ETE. If at x we have exactly k agents with $x_i = a$ for some a, order them arbitrarily, say $i_1 < i_2 < ... < i_k$, and perform the transfers

$$t_{i_1} = \frac{k-1}{2}a, t_{i_2} = \frac{k-3}{2}a, ...t_{i_k} = -\frac{(k-1)}{2}a$$

In other words, the mechanism is defined up to a tie-breaking rule, but the corresponding method is unique:

$$y_{i_1} = y_{i_2} = ... = y_{i_k} = \frac{k+1}{2}a + \sum_{j:x_j<a} x_j$$

Our next two examples are efficient scheduling methods meeting ETE and CONT, namely the *proportional* method:

$$y_i = \frac{x_i}{x_N} \cdot v(N,x) \text{ for all } x \neq 0; y = 0 \text{ for } x = 0, \quad (2)$$

and the *egalitarian* method:

$$y_i = x_i + \frac{1}{n}(\sum_{N(2)} x_i \wedge x_j) \text{ for all } N, x. \tag{3}$$

The latter charges the same net cost to every agent beyond his/her own stand alone cost. The proportional method meets all (and the egaliatrian method meets some) equity tests discussed in Section 5, yet both are vulnerable to the coalitional maneuvers to which we now turn.

4 Merging and Splitting

The server can recognize the length of the jobs it performs, but not the identity of the beneficiary of those jobs. We describe first the merging of several jobs under a single identity, then turn to splitting a given job in several small jobs under multiple identities.

Given $N \subseteq \mathcal{N}$, a coalition S, $S \subseteq N$, and an agent $i^* \in S$, we associate to every problem (N, x) the $(S, i^*)-$*merged problem* (N^*, x^*) as follows

$$N^* = (N \backslash S) \cup \{i^*\}; x^*_{i^*} = x_S \text{ and } x^*_j = x_j \text{ for all } j \in N \backslash S$$

We also use the notation $v(S, x) = x_S + \sum_{S(2)} x_i \wedge x_j$ for the stand alone waiting cost of coalition S, namely the efficient total wait of S when it is served before $N \backslash S$. Given a mechanism μ on $\mathcal{N}$ we define:

Merge-proofness (MPF): for all N, S, i^* as above and all $x \in \mathbb{R}^N_+$:

$$\mu(N^*, x^*) = (\sigma^*, t^*) \Rightarrow y_S(N, x) \leq v(S, x) + k(S, x) \cdot x_{P(i^*, \sigma^*)} + t^*_{i^*} \tag{4}$$

where $k(S, x)$ is the number of agents $i \in S$ such that $x_i > 0$.
In this inequality the left-hand side is the net waiting cost of coalition S before merging, and the right-hand side its net cost after merging. Indeed coalition S uses efficiently the slot of length x_S allocated to agent i^*, and moreover everyone in S with a non null job must wait until completion of all jobs in $P(i^*, \sigma^*)$. Note that for $S = N$, the merge-proofness inequality is just the efficiency property.

Given $N \subseteq \mathcal{N}, i_* \in N$, and a finite set $T \subseteq \mathcal{N}, T \cap N = \varnothing$, we associate to every problem (N, x), the family of $(T, i_*)-$*splitted problems* (N_*, x_*) as follows

$$N_* = N \cup T; (x_*)_{T \cup i_*} = x_{i_*} \text{ and } (x_*)_j = x_j \text{ for all } j \in N \diagdown i_*$$

Given a mechanism μ on $\mathcal{N}$ we define:

Split-proofness (SPF): for all N, T, i_* as above, all $x \in \mathbb{R}_+^N$ and all (T, i_*)-splitted problem (N_*, x_*)

$$\mu(N_*, x_*) = (\sigma_*, t_*) \Rightarrow y_{i_*}(N, x) \leq x_{i_*} + x_{P(j_*, \sigma_*)} + (t_*)_{T \cup i_*} \tag{5}$$

where j_* is the last agent in $T \cup i_*$ for σ_*.
Agent i_*'s net cost before splitting is on the left-hand side; after the split, i_* must wait until all jobs in $P(j_*, \sigma_*)$ are completed[2], therefore the right-hand side is his net cost.

As a first application of these definitions, it is easy to verify that Shortest Job First is not split-proof, but it is merge-proof. Symmetrically, Longest Job First is split-proof, but not merge-proof.

We check now that the egalitarian method (3) is neither merge-proof nor split-proof. In the problem $N = \{1, 2, 3\}, x = (1, 1, 4)$, consider the split of x_3 into x_{*3} and x_{*4}, with $x_{*3} = x_{*4} = 2$. The *actual* wait of agent 3 in (N, x) is the same as in $N_* = \{1, 2, 3, 4\}, x_* = (1, 1, 2, 2)-$ under the assumed identities. The split is profitable because the monetary transfer is smaller in the latter:

$$y_3 = 6 + t_3 = 4 + \frac{1}{3}(1 + 1 + 1) \Rightarrow t_3 = -1$$

$$(y_*)_{34} = 4 + 6 + (t_*)_{34} = 2(2 + \frac{1}{4}(7)) \Rightarrow (t_*)_{34} = -2.5$$

Next consider the problem $N = \{1^*, 2, 3, 4\}, x = (2, 2, 5, 5)$. The actual total wait of agents 1,2 in (N, x) is unchanged as they merge to 1^* in $N^* = \{1^*, 3, 4\}, x^* = (4, 5, 5)$. The move is profitable because the net transfer decreases:

$$y_{1^*2} = 6 + t_{12} = 2(2 + \frac{1}{4}(15)) \Rightarrow t_{12} = 5.5$$

$$y^*_{1^*} = 4 + t^*_{1^*} = 4 + \frac{1}{3}(13) \Rightarrow t^*_{1^*} = 4.33$$

We let the reader check similarly that the proportional method (2) is not split-proof, by considering the split of agent 1 from $(\{1, 2\}, (5, 4))$ into $(\{1, 2, 3\}, (5, 2, 2))$.

[2] At least if $(x_*)_{j_*} > 0$. If $(x_*)_{j_*} = 0$, we should replace j_* by the last agent in $T \cup i_*$ with a positive job. But this does not affect the statement of SPF.

On the other hand, the proportional method is merge-proof. We omit the easy proof.

Theorem 1 *Assume* $|N| \geq 4$. *There is no scheduling mechanism satisfying Merge-proofness, Split-proofness, and either Continuity or Equal Treatment of Equals.*

The proof is in the Appendix. Recall that merge-proofness implies in particular efficiency. If we restrict the merge-proofness property by allowing only the merging of proper coalitions, there may exist some (inefficient) mechanisms meeting MPF and SPF. I conjecture that this is not the case.

5 Unpalatable consequences of Split-proofness

The formal similarity between merging and splitting suggests that the properties MPF and SPF are comparably demanding. This intuition is not correct. We list below five mild normative requirements that we may want to impose on a scheduling method. Then we show that any "reasonable" split-proof method must violate each one of these four properties. In the following statements, we fix a method $(N, x) \to y$:

- **Monotonicity** (MON): $x_i \to y_i(N, x)$ is non-decreasing, for all N, i, $x_{-i} \in \mathbb{R}_+^{N \setminus i}$
- **Ranking** (RKG): $\{x_i \leq x_j\} \Rightarrow \{y_i \leq y_j\}$ for all N, i, j, x
- **Stand Alone Bound** (SAB): $y_i \geq x_i$ for all N, i, x
- **Zero Charge for Null Jobs** (ZCNJ): $x_i = 0 \Rightarrow y_i = 0$ for all N, i, x
- **Finite Liability** (FL): $\sup_{x_{-i} \in \mathbb{R}_+^{N \setminus i}} y_i(N, (x_i, x_{-i})) < +\infty$ for all N, i, x_i

The first three properties are standard equity tests. The Stand Alone Bound sets a minimal net waiting cost, namely my disutility in the most optimistic case where I have absolute priority for service. It rules out the subsidization of any agent beyond this most advantageous situation. Monotonicity says that my net waiting cost weakly increases when the service time of my job increases: besides its clear normative meaning, this property also rules out "sabotage" by artificially increasing one's job size. Ranking conveys a related idea by way of interpersonal comparisons: if my job is larger than yours, my responsibility in the total waiting burden is higher.

Zero Charge for Null Jobs frees a "null job" agent of any responsibility: such an agent is served first by efficiency, and causes no additional waiting cost to any one: the axiom says that he should not be taxed either, $t_i = 0$. The combination of Continuity and Zero Charge for Null Jobs implies that y_i converges to zero with x_i.

Finally, Finite Liability prevents a job of a given size to pay arbitrary large fees when other jobs become very large. It encourages participation of a risk averse user who has no information about other jobs' sizes.

Many scheduling methods meet these five properties. Examples include Shortest Job First (example 1) and the proportional method (2): check that $\sup_{x_{-i}} y_i(N, (x_i, x_{-i})) = n \cdot x_i$ for the former, and $= \frac{(n+1)}{2} \cdot x_i$ for the latter. Moreover, all five properties are preserved under convex combinations. The egalitarian method (3) fails ZCNJ and FL, but meets the other three.

We now state a negative result about split-proof scheduling methods.

Proposition 1 *Fix N and an efficient and continous scheduling method m treating equals equally. If m is split-proof and $\mathcal{N} \geq 5$ then it must fails Monotonicity, Ranking and the Stand Alone bound. If $|\mathcal{N}| \geq \infty$, it fails Zero Charge for Null Jobs and Finite Liability as well.*

By contrast the S^+ solution defined in the next Section, is an efficient merge-proof method meeting all the other axioms discussed so far: ETE, CONT, MON, RKG, SAB, ZCNJ and FL.

Proof

Monotonicity. Let $N = \{1,2,3,4\}$ and $x(\varepsilon) = (1,1,1,2(1+\varepsilon))$. Consider the split of agent 4 into agents 4,5 and $x_*(\varepsilon) = (1,1,1,1+\varepsilon,1+\varepsilon)$. By CONT and ETE:

$$\lim_{\varepsilon \to 0} y(x_*(\varepsilon)) = y(x_*(0)) = (3,3,3,3,3)$$

By efficiency and CONT again:

$$\{y_{45}(x_*(\varepsilon)) = 4 + \varepsilon + 5 + \varepsilon + t_{45}(x_*(\varepsilon)) \to 6\} \Rightarrow \lim_{\varepsilon \to 0} t_{45}(x_*(\varepsilon)) = -3$$

Because the split is not profitable for agent 4, and her real wait after the split is unchanged, we have

$$y_4(x(\varepsilon)) \leq 5 + 2\varepsilon + t_{45}(x_*(\varepsilon)) \Rightarrow y_4(x(0)) \leq 2$$

On the other hand at $\overline{x} = (1,1,1,1)$ ETE gives $y_4(\overline{x}) = 2.5$, and we see that Monotonicity is violated as x_4 goes from 1 to 2.

Ranking. Let $N = \{1,2,3\}$ and $x(\varepsilon) = (1,1,2(1+\varepsilon))$. Consider the split of agent 3 into 3,4 and $x_*(\varepsilon) = (1,1,1+\varepsilon,1+\varepsilon)$. Mimicking the argument of the proof above we get successively

$$\lim_{\varepsilon\to 0} y(x_*(\varepsilon)) = (2.5,2.5,2.5,2.5), \lim_{\varepsilon\to 0} t_{34}(x_*(\varepsilon)) = -2, \text{ and } y_3(x(0)) \leq 2$$

Now efficiency and ETE give $y_1(x(0)) = y_2(x(0)) \geq 2.5$, a contradiction of RKG.
Stand Alone bound. Let $N = \{1,2\}$and $x(\varepsilon) = (1,3(1+\varepsilon))$. Consider the split of agent 2 into 2,3,4 and $x_*(\varepsilon) = (1,1+\varepsilon,1+\varepsilon,1+\varepsilon)$. As before we have successively

$$\begin{aligned}\lim_{\varepsilon\to 0} y(x_*(\varepsilon)) &= (2.5,2.5,2.5,2.5), \lim_{\varepsilon\to 0} t_{234}(x_*(\varepsilon)) = -1.5,\\ &\Rightarrow \quad y_2(x(0)) \leq 2.5 < 3 = x_2(0)\end{aligned}$$

This contradicts SAB.
Zero Charge for Null Jobs. Let $N = \{1,2\}$ and fix an integer $p, p \geq 2$. Set $x(\varepsilon) = (\frac{1}{p}, 1+p\varepsilon)$ and consider the split of 2 into $2,3,...,p+1$ and $x_*(\varepsilon) = (\frac{1}{p}, \frac{1}{p}+\varepsilon, ..., \frac{1}{p}+\varepsilon)$. As before we have $\lim_{\varepsilon\to 0} y_i(x_*(\varepsilon)) = \frac{p+2}{2p}$ implying $\lim_{\varepsilon\to 0} t_1(x_*(\varepsilon)) = \frac{1}{2}$. Then Split-proofness implies $y_2(x(\varepsilon)) \leq 1+p\varepsilon+\frac{1}{p} - t_1(x_*(\varepsilon))$ hence $y_2(x(0)) \leq \frac{1}{2}+\frac{1}{p} \Longleftrightarrow y_1(x(0)) \geq \frac{1}{2}+\frac{1}{p}$. But CONT and ZCNJ imply $\lim_{p\to\infty} y_1((\frac{1}{p},1)) = 0$, contradiction.
Finite Liability. Set $N = \{1,2\}$ and $x(\varepsilon) = (a, k\cdot(a+\varepsilon))$, where $a > 0$ is arbitrary. Consider the split of 2 to $2,3,..,k+1$ in $x_*(\varepsilon) = (a, a+\varepsilon, .., a+\varepsilon)$. We have $\lim_{\varepsilon\to 0} t_{\{2,..,k+1\}}(x_*(\varepsilon)) = -\frac{k}{2}\cdot a$, and Split-proofness implies

$$y_2(x(\varepsilon)) \leq a + k\cdot(a+\varepsilon) + t_{\{2,..,k+1\}}(x_*(\varepsilon)) \Longrightarrow y_1(x(0)) \geq a + \frac{k}{2}\cdot a$$

from which $\sup_k y_1(a, k\cdot a)$ follows. This proves the claim when $|N| = 2$. The argument is similar for an arbitrary number of agents.

Remark 1 For the statements about Ranking and the Stand Alone bound, the assumption Equal Treatment of Equals is redundant. In other words, any efficient, continous and split-proof method must violate Ranking and the Stand Alone bound for $|N| \geq 4$. To check this, take a set N_* with four agents. Setting $x_*(0) = (1,1,1,1)$, we have $y_{N_*}(x_*(0)) = 10$ thus there exists a pair i,j in N_* such that $y_{ij}((1,1,1,1)) \leq 5$. Label the agents so that $i = 3, j = 4$ and $N_* = \{1, 2, 3, 4\}$. Define $N = \{1, 2, 3\}, x(\varepsilon)$ and $x_*(\varepsilon)$, as in the above argument about Ranking. Continuity ensures $\lim_{\varepsilon \to 0} y_{34}(x_*(\varepsilon)) \leq 5$, then $\lim_{\varepsilon \to 0} t_{34}(x_*(\varepsilon)) \leq -2$. Split-proofness applied to the split of 3 in $(N, x(\varepsilon))$ to $3, 4$ in $(N_*, x_*(\varepsilon))$ gives $y_3(x(0)) \leq 2$. Therefore $y_{12}(x(0)) \geq 5$ so that $y_i(x(0)) \geq 2.5$ for at least one of 1,2. Thus Ranking fails. The similar proof for the Stand Alone Bound is omitted for brevity.

Whether or not we can drop Equal Treatment from the assumptions in the two remaining statements is an open question.

6 Separable scheduling methods

The total waiting externality in the problem (N, x) is $v(N, x) = x_N + \sum_{N(2)} x_i \wedge x_j$, namely the cost of having to share the server. A *separable* method shares each pairwise externality $x_i \wedge x_j$ independently of the rest of the jobs.

Definition 1 *Choose a continuous function θ from $\mathbb{R}^2_+$ into $\mathbb{R}$ such that $\theta(a, b) + \theta(b, a) = a \wedge b$ for all $a, b \in \mathbb{R}_+$. The θ-separable scheduling method is given by*

$$y_i(N, x) = x_i + \sum_{N \setminus i} \theta(x_i, x_j) \text{ for all } N, i \in N \text{ and } x \in \mathbb{R}^N_+$$

The θ-separable method is obviously efficient, continous, and treats equals equally.

The Shortest Job First method is θ-separable, except that the function θ is not continuous:

$$\theta(a, b) = 0 \text{ if } a < b; \theta(a, b) = \frac{a}{2} \text{ if } a = b; \theta(a, b) = a \text{ if } a > b$$

Neither the egalitarian nor the proportional method is separable.

We speak of a $\theta-$separable mechanism for any mechanism generating the method in Definition 1.

Proposition 2

a) *The θ-separable scheduling method meets*

i) Monotonicity iff $\{\theta(a, b)$ is non-decreasing in $a\}$.

ii) Ranking iff $\{\theta(a, b)$ is non-decreasing in a and $a \leq b \Rightarrow \theta(a, b) \leq \frac{b}{2}\}$.

iii) Stand Alone Bound iff $\{\theta(a, b) \geq 0$ for all $a, b\}$.

iv) Zero Charge for Null Job iff $\{\theta(0, b) = 0$ for all $b\}$.

v) Finite Liability iff $\sup_b \theta(a,b) < +\infty$ *for all* a.
b) *A* θ*-separable mechanism is merge-proof iff*

$$\begin{aligned} \theta(a_1,b)+\theta(a_2,b) &\leq \theta(a_1+a_2,b) \text{ for all } b,a_1,a_2 \; : a_1+a_2 \leq b \\ \theta(a_1,b)+\theta(a_2,b) &\leq \theta(a_1+a_2,b)+b \text{ for all } b,a_1,a_2 \end{aligned} \tag{6}$$

c) *A* θ*-separable mechanism is split-proof iff*

$$\begin{aligned} \theta(a_1+a_2,b)+b &\leq \theta(a_1,b)+\theta(a_2,b) \text{ for all } b,a_1,a_2 : b \leq a_1,a_2 \\ \theta(a_1+a_2,b) &\leq \theta(a_1,b)+\theta(a_2,b) \text{ for all } b,a_1,a_2 \end{aligned} \tag{7}$$

Proof
Statement i. Suppose $\theta(a,b) > \theta(a',b)$ for some $a < a'$. Fix n and consider the $(n+1)-$agents profiles $x = (a,b,...b)$ and $x' = (a',b,..,b)$. For n large enough, we have

$$y_1(x) = a + n\cdot\theta(a,b) > a' + n\cdot\theta(a',b) = y_1(x')$$

contradicting MON. The converse statement is obvious.
Statement ii. Suppose $\theta(a,b) > \theta(a',b)$ for some $a < a'$. Consider the $(n+2)-$agents profile $x = (a,a',b,...b)$. For n large enough we get $y_1 > y_2$, contradicting Ranking. Thus θ must be monotonic in its first variable. Next we fix $a,b,a \leq b$, and apply Ranking to $x = (a,b)$:

$$y_1(x) = a + \theta(a,b) \leq y_2(x) = b + \theta(b,a) = b + a - \theta(a,b)$$

establishing the second property in statement *ii*. The converse property is just as easy.

Merge-proofness. Fix N,S,i^*,x as in the premises of (4) and develop this inequality for our $\theta-$separable method. Compute first $t^*_{i^*}$:

$$y_{i^*}(N^*,x^*) = x^*_{i^*} + x^*_{P(i^*,\sigma^*)} + t^*_{i^*} = x^*_{i^*} + \sum_{N\backslash S}\theta(x^*_{i^*},x_j)$$

$$\Longrightarrow t^*_{i^*} = -x_{P(i^*,\sigma^*)} + \sum_{N\backslash S}\theta(x_S,x_j)$$

Next the definition of θ implies

$$\begin{aligned} y_S(N,x) &= x_S + \sum_{S(2)} \{\theta(x_i,x_j)+\theta(x_j,x_i)\} + \sum_{i\in S, j\in N\setminus S} \theta(x_i,x_j) \\ &= v(S,x) + \sum_{j\in N\setminus S}\sum_{i\in S}\theta(x_i,x_j) \end{aligned}$$

Therefore inequality (4) amounts to

$$0 \leq (k(S,x)-1)\cdot x_{P(i^*,\sigma^*)} + \sum_{j\in N\setminus S}\{\theta(x_S,x_j) - \sum_{i\in S}\theta(x_i,x_j)\} \tag{8}$$

Check that if θ satisfies (6), then (8) holds for all N, S, i^* and x. The top inequality in (6) implies $\theta(0,b)\leq 0$. Repeated applications of the bottom one give

$$\sum_S \theta(x_i,b) \leq \theta(x_S,b) + (k(S,x)-1)\cdot b, \text{ and}$$

$$\sum_S \theta(x_i,b) \leq \theta(x_S,b) \text{ if } x_S \leq b$$

Applying the top inequality to $b=x_j$ for all $j\in P(i^*,\sigma^*)$, and the bottom one to x_j for all $j\in N\setminus(S\cup P(i^*,\sigma^*))$ gives the desired inequality (8).
Next we prove that (6) must hold if θ meets (8) for all problems and all merging. Consider $N=\{1,2,3\}, S=\{1,2\}, i^*=1$ and $x=(a_1,a_2,b)$ for arbitrary a_i, b in $\mathbb{R}_+$. If $a_1+a_2<b, P(i^*,\sigma^*)$ is empty and (8) yields the top inequality in (6). Continuity of θ takes care of the case $a_1+a_2=b$. If $a_1+a_2>b, P(i^*,\sigma^*)=\{3\}$ and (8) gives the bottom inequality in (6).

Split-proofness. We develop similarly inequality (5) for the θ-separable method. First we compute $(t_*)_{T\cup i_*}$ in (5). Set $T\cup i^*=R$ and $|R|=r$, then relabel agents in R as $1,2,..,r$ with $(x_*)_1\geq(x_*)_2\geq ... \geq (x_*)_r$. Thus $x_{i_*}=\sum_1^r(x_*)_k$. In the split problem (N_*,x_*), the total wait of coalition R (if this coalition was truly made of r different agents) is $v(R,x_*)+\sum_{k=1}^r k\cdot x_{S_k}$ where S_k contains those agents in $N\setminus i_*$ ranked before k and after $k+1$ in σ_*. In particular for $j\in S_k, (x_*)_k\geq x_j\geq(x_*)_{k+1}$. We can now compute the net cost of R in two ways:

$$y_R(N_*,x_*) = v(R,x_*) + \sum_1^r k\cdot x_{S_k} + (t_*)_R$$

$$= v(R, x_*) + \sum_{j \in N \setminus i_*} \sum_1^r \theta((x_*)_k, x_j)$$

from which we get $(t_*)_R$. Substituting in the split-proofness inequality (5) we get

$$x_{i_*} + \sum_{j \in N \setminus i_*} \theta(x_{i_*}, x_j) \leq x_{i_*} + \sum_1^r x_{S_k} + (t_*)_R$$

$$\iff \sum_{j \in N \setminus i_*} \theta(x_{i_*}, x_j) + \sum_1^r (k-1) \cdot x_{S_k} \leq \sum_{j \in N \setminus i_*} \sum_1^r \theta((x_*)_k, x_j) \qquad (9)$$

We show finally that (7) is true if and only if (9) holds for all N, i_*, T and x. The "if" statement follows easily from applying (9) to $N = \{1, 2\}, i_* = 2, T = \{3\}, x = (b, a_1 + a_2)$ and $x_* = (b, a_1, a_2)$. If $b \leq a_1, a_2$ we have $S_2 = \{1\}$ and we get the top inequality in (7). For other values of b, S_2 is empty and we get the rest of (7).

Before proving the "only if" statement, we notice a consequence of (7). Fix k, $a_k, k = 1, .., r$ such that $a_1 \geq a_2 \geq .... \geq a_r$, and b. We have

$$b \leq a_k \Longrightarrow \theta(a_1 + .. + a_r, b) + (k-1) \cdot b \leq \sum_{s=1}^r \theta(a_s, b)$$

We omit the easy proof. Apply this inequality to $a_k = (x_*)_k$ and to $b = x_j$ for some agent j in S_k, we get

$$\theta(x_{i_*}, x_j) + (k-1) \cdot x_j \leq \sum_{s=1}^r \theta((x_*)_s, x_j)$$

Summing up over all $j \in N \setminus i_*$ gives (9) as desired.

Proposition 2 shows that among separable scheduling methods, it is easy to ensure merge-proofness or split-proofness. The former requires θ to be something less than superadditive in its first variable; the latter requires θ to be something more than subadditive in its first variable. The two requirements are incompatible: this results from Theorem 1, or can be checked directly by comparing systems (6) and (7).

Two separable methods stand out for the simplicity of their definition and their multiple interpretations. Moreover, they are the backbone of the characterization of transfer-proof methods in the next section.

Definition 2 *The S^+ and S^- separable methods are associated with θ^+ and θ^- respectively.*

$$\theta^+(a,b) = \frac{1}{2}(a \wedge b);\ \theta^-(a,b) = b - \frac{1}{2}(a \vee b) \text{ for all } a,b$$

The corresponding net waiting costs and transfers for a problem (N,x) with $|N| = n$ and $x_1 \leq x_2 \leq ... \leq x_n$ are:

$$y_i^+ = \frac{1}{2}x_{\{1,i-1\}} + (1 + \frac{n-i}{2})x_i \text{ and } t_i^+ = \frac{1}{2}((n-i)x_i - x_{\{1,i-1\}})$$

$$y_i^- = x_{\{1,i-1\}} - (\frac{i-3}{2})x_i + \frac{1}{2}x_{\{i+1,n\}} \text{ and } t_i^- = \frac{1}{2}(x_{\{i+1,n\}} - (i-1)x_i)$$

where we use the notation $x_{\{i,j\}} = \sum_{i \leq k \leq j} x_k$.
These formulas follow easily from Definition 1.

The S^+ method divides equally the externality $x_i \wedge x_j$ between x_i and x_j. If $x_i < x_j$, agent i is served first and gives a "rebate" $\frac{1}{2}x_i$ to agent j. With the S^- method, agent i gives a larger rebate $\frac{1}{2}x_j$.

Notice that for $|N| = 2$, S^- simply equalizes net costs $y_1^- = y_2^- = x_1 + \frac{1}{2}x_2$, a fairly reasonable compromise. But for larger sizes of N, the method S^- has several unappealing features.

Proposition 3
i) The scheduling method S^+ is merge-proof. It also satisfies Monotonicity, Ranking, Stand Alone Bound, Zero Charge for Null Jobs and Finite Liability.
ii) The scheduling method S^- is split-proof. Hence it violates these five properties.

Proof
That S^+ meets the five properties MON, RKG, SAB, ZCNJ and FL is obvious, either by direct inspection of the formula for y_i^+, or by invoking Proposition 2. Note that the liability of job i among the users N with $|N| = n$ is $\sup_{x_{-i}} y_i((x_i, x_{-i})) = \frac{n+1}{2} \cdot x_i$. This is the smallest feasible liability in our model.

Proposition 1 and Split-proofness imply that S^- violates all five properties; this fact can also be checked directly on the formula for y_i^-, or by invoking Proposition 2. In particular S^- has the following "anti-ranking" property: $x_i < x_j \Longrightarrow y_i > y_j$.
Next one checks easily that the function θ^+ has the subadditivity properties (6), whereas θ^- has the superadditivity properties (7), and the proof is complete.

We conclude this section with several alternative interpretations of S^+ and S^-.

Lemma 1 *The profile of net costs selected by the method S^+ is the Shapley value of the optimistic Stand Alone cooperative game $S \to v(S,x)$ for all $S \subseteq N$. The profile selected by the method S^- is the Shapley value of the pessimistic stand alone game $S \to w(S,x) = |S| \cdot x_{N \setminus S} + v(S,x)$.*

In the optimistic (resp. pessimistic) Stand Alone game, the total cost of a coalition S is its efficient cost when it is served before (resp. after) the complement coalition $N \diagdown S$.

Proof. The interpretation of S^+ as the Shapley value of the optimistic game is already in Curiel et al [2002]. For the sake of completeness, we give a proof here. Given $N, S \subseteq N$ and $i \in N \setminus S$, the marginal contribution of agent i to S is

$$v(S \cup i, x) - v(S,x) = x_i + \sum_{j \in S} x_i \wedge x_j$$

Therefore the $(i,j)-$externality $x_i \wedge x_j$ is charged to agent i if and only if j appears before i in the random ordering of N: this happens with probability .5, so the Shapley value awards precisely y_i^+ to agent i.

Next we check that y^- is the Shapley value of the game w. By additivity of the value this amounts to check that $y^- - y^+$ is the value of the game $\alpha = w - v$. Compute:

$$y_i^- - y_i^+ = \frac{1}{2} x_N - \frac{n}{2} x_i \text{ and } \alpha(S \cup i, x) - \alpha(S,x) = x_{N \setminus S \setminus i} - |S| \cdot x_i$$

from which the desired conclusion follows easily.

Remark 2 Yet another interpretation of S^+ is by means of the serial cost sharing formula of Friedman and Moulin [1999]. Consider the scheduling problem (N,x) as a cost sharing problem with the demand profile x and the cost function $C(x) = v(N,x)$. One checks easily that y^+ is the profile of cost shares under the serial cost sharing formula defined there. Finally, we note that under S^+, the transfer t_i^+ to agent i does not depend upon the length of jobs longer than x_i; whereas under S^-, t_i^- is independent of the length of jobs shorter than x_i. In combination with efficiency and equal treatment of equals, these properties are clearly characteristic. In the related scheduling model where all jobs are of equal length but agents differ by their linear waiting cost, Maniquet [2003] and Chun [2004] use similar independence properties to characterize respectively the analog of our S^+ and S^- scheduling methods.

7 Transfer of jobs and the main result

We consider a manipulation related to merging and splitting, yet more subtle because it involves a partial transfer of jobs. The number of agents remains constant during the transfer, therefore in this section we may assume $N = \mathcal{N}$.

Our main result (Theorem 2 below) characterizes the scheduling mechanisms robust against partial transfers of jobs involving only *two* agents, together with monetary transfers among possibly more agents. This restriction is crucial. In Section 8 we derive an impossibility result when transfers among three agents or more are feasible.

Given $N, i, j \in N, \varepsilon > 0$, and two profiles x, x' such that $x_k = x'_k$ for $k \in N \diagdown \{i, j\}$, we call x' an $\varepsilon-$*shrink* (resp. an $\varepsilon-$*spread)* of x by i, j if

$$x'_i \leq x'_j \Leftrightarrow x_i \leq x_j;\ x'_i + x'_j = x_i + x_j;\ |x'_i - x'_j| = |x_i - x_j| - 2\varepsilon \text{ (resp. } |x'_i - x'_j| = |x_i - x_j| + 2\varepsilon)$$

Finally the notation $\Delta(\sigma; i, j)$ stands for the set of agents in N that ordering σ ranks between i and j. We are now ready to define the two sides of the transfer-proofness axiom. Throughout these definitions we fix the set N of agents, $|N| \geq 3$.

Pairwise Shrink-proofness: for all $S, S \subset N$, x, and $\varepsilon-$shrink x' of x by $i, j \in S$

$$y_S(N, x) \leq y_S(N, x') - \varepsilon \tag{10}$$

Pairwise Spread-proofness: for all S,$S \subset N, x$, and $\varepsilon-$spread x' of x by $i, j \in S$

$$\sigma' = \mu(N, x') \Rightarrow y_S(N, x) \leq y_S(N, x') + \varepsilon + x_{\Delta(\sigma'; i, j)} \tag{11}$$

Note that we do not allow pairwise transfers exchanging the ordering of jobs 1 and 2, as when x_1, x_2 with $x_1 < x_2$ becomes x'_1, x'_2 with $x_1 \leq x'_2 \leq x'_1 \leq x_2$ and $x'_1 + x'_2 = x_1 + x_2$. This restriction is without any loss of generality, because the deviating agents have every incentive to use efficiently the time slots allocated to their reported jobs. In the configuration above, the slot for x'_2 will be used to complete job x_1 and start job x_2. Therefore the shift from x to x' is equivalent to a shrink from (x_1, x_2) to (x'_2, x'_1).

Definition 3 *We call the mechanism μ* ***pairwise transfer-proof*** *(PTP) if it is pairwise shrink-proof and spread-proof.*

Several comments on this definition are in order. Firstly, the PTP concept applies to scheduling mechanisms because, as explained below, the choice of

$\sigma' = \mu(N, x')$ matters to the spread-proofness property (but not to that of pairwise shrink-proofness).

The second observation is that PTP rules out certain maneuvers by coalitions S of arbitrary size: although the partial transfer of jobs only concerns two agents, other agents in S are involved in cash transfers inside S.

Next we comment on the inequality defining shrink-proofness. The left-hand side is the total net cost of coalition S before the (job and cash) transfers. We claim that the right-hand side is its total net cost after the job transfer. Without loss of generality, suppose $i = 1, j = 2$ and $x'_1 = x_1 + \varepsilon \leq x_2 - \varepsilon = x'_2$. The real job x_1 will be completed whenever x'_1 is served, and job x_2 when x'_2 is served. If a reported job x'_1 or x'_2 is served after some agent $j, j \neq 1, 2$, so does the corresponding real job, and vice-versa. Thus the difference between the waiting time of the real jobs x_1, x_2, and that of the reported jobs x_1, x_2 is $2x_1 + x_2 - (2x'_1 + x'_2) = -\varepsilon$. Hence inequality (10).

For instance, we check that the proportional mechanism is not pairwise shrink-proof. Let $N = \{1, 2, 3\}, x = (1, 6, 5)$ and $S = \{1, 2\}$ with $x'_1 = 3, x'_2 = 4$. Thus x' is a 2-shrink of x by $1, 2$, involving no other agent. Compute

$$y_{12}(x) = \frac{7}{12} \cdot (10) > \frac{7}{12} \cdot (22) - 2 = y_{12}(x') - \varepsilon$$

Recall that this method is in fact merge-proof. We let the reader check that the egalitarian method also fails (10) for the following three-person example:

$$x = (1, 8, 2), x' = (4, 5, 2) \Rightarrow y_{12}(x) = 11\frac{2}{3}, y_{12}(x') = 14\frac{1}{3}$$

Finally we explain inequality (11). Suppose $i = 1, j = 2, x'_1 = x_1 - \varepsilon, x'_2 = x_2 + \varepsilon, x_1 \leq x_2$. After the report, the real job x_1 will not be completed when job x'_1 is done, but only during the service of job x'_2. Thus the difference between the wait of the real jobs and that of the reported jobs is

$$2x_1 + x_2 + x_{\Delta(\sigma';1,2)} - (2x'_1 + x'_2) = \varepsilon + x_{\Delta(\sigma';1,2)}$$

If the set $\Delta(\sigma'; 1, 2)$ is not empty, a spread from x to x' introduces the additional waiting time $x_{\Delta(\sigma';1,2)}$ to the reported waiting time of S at x'. Thus pairwise spread-proofness ends up being easier to meet than pairwise shrink-proofness. For instance, all three methods Shortest Job First, proportional and egalitarian are spread-proof.

For an example where pairwise spread-proofness is violated, consider the following $\theta-$separable method:

$$\theta(a,b) = \frac{ab}{a+b} \text{ if } a \leq b; = \frac{b^2}{a+b} \text{ if } b \leq a$$

Set $N = \{1,2,3\}$ and $x = (1,2,3)$. Consider the $\varepsilon-$spread by $\{1,2\}$ to $x' = (1-\varepsilon, 2+\varepsilon, 3)$ with $0 < \varepsilon < 1$. Inequality (11) for $S = \{1,2\}$ reads

$$y_{12}(x) = 4 + \theta(1,3) + \theta(2,3) \leq 4 + \theta(1-\varepsilon,3) + \theta(2+\varepsilon,3) = y_{12}(x') + \varepsilon$$

It is violated because $\theta(a,b)$ is strictly concave in a on $[0,b]$.

We are ready to state our main characterization result.

Theorem 2 *Fix N with $|N| \geq 4$.*
i) Choose two continuous functions, $\alpha : \mathbb{R}_+ \to \mathbb{R}$ and $\gamma : \mathbb{R}_+ \to \mathbb{R}^N$ such that $\sum_N \gamma_i(z) = 0$ for all z. Define a scheduling method y as follows:

$$y(x) = \alpha(x_N) \cdot y^+(x) + (1-\alpha(x_N)) \cdot y^-(x) + \gamma(x_N) \text{ for all } x \in \mathbb{R}_+^N$$

where y^+, y^- are the methods S^+ and S^- as in Definition 2. Any corresponding mechanism is efficient, continuous, and pairwise transfer-proof.
ii) Conversely, if a mechanism μ is efficient, continuous and pairwise transferproof, the associated method y takes the above form.
The proof is in the Appendix.

The PTP axiom, almost single handedly, captures a fairly small family of scheduling methods/mechanisms. This family contains the affine combinations of S^+, S^- to which we can add some "constant" γ, where the coefficients of the affine combination and the constant depend only upon x_N.

The mild property of Scale Invariance allows us to pick a one-dimensional family of methods:
Scale Invariance (SI): $y(\lambda x) = \lambda y(x)$ for all $\lambda > 0, x \in \mathbb{R}_+^N$

Corollary 1 to Theorem 2 *Consider a mechanism μ defined as in statement $i)$ by the functions α and γ :*
i) μ treats equals equally iff $\gamma(z) = 0$ for all z.
ii) μ is scale invariant iff α is constant and γ is homogeneous of degree 1.

If we combine Continuity, Equal Treatment, Scale Invariance, with efficiency and Pairwise Transfer-proofness, Corollary 1 tells us that we are left with the one-dimensional line of methods joining S^+ and S^-. These methods are all separable, with corresponding function θ :

$$\theta(a,b) = \frac{1}{2}(a \wedge b) - \frac{(1-\alpha)}{2}(a-b) \text{ for all } a,b \geq 0$$

The parameter α is any *real* number. The method S^+ obtains for $\alpha = 1$ and S^- for $\alpha = 0$.

Corollary 2 to Theorem 2 *Consider a mechanism μ defined as in statement i) by the functions α and γ :*
i) μ is merge-proof if and only if $\alpha(z) \geq 1$ for all z.
ii) μ is split-proof if and only if $\alpha(z) \leq 0$ for all z.

This establishes the polar role of S^+ and S^- within the family described in Theorem 2, or in Corollary 1.

If Theorem 2 and its Corollaries 1 and 2 give a symmetrical role to S^+ and S^-, this symmetry is destroyed as soon as we introduce the normative requirements of Section 5. Not surprisingly, these properties point toward the method S^+.

Corollary 3 to Theorem 2 *The mechanism S^+ is characterized by the combination of efficiency, Continuity, Pairwise Transfer-proofness and either Zero Charge for Null Jobs, or the Stand Alone bound.*

Remark 3 Two additional properties can be used to single out the S^+ method. They both place an upper bound on individual net waiting costs, which is the familiar idea of a lower bound on individual welfare. The *pessimistic stand alone bound* for agent i is simply $y_i \leq w(\{i\}, x) = x_N$. Both S^+ and S^- meet this bound. The *unanimity bound* for agent i is $y_i \leq \frac{n+1}{2} x_i$. It is this agent's net cost in a hypothetical problem $(N, \widetilde{x})$ where all jobs are of size x_i. As indicated in the proof of proposition 3, S^+ meets this bound, whereas S^- violates it, even for $|N| = 2$.

Now S^+ is characterized by the combination of efficiency, Continuity, Pairwise Transfer-proofness and *either* {the pessimistic stand alone bound plus the unanimity bound}, or {the pessimistic stand alone bound and merge-proofness}. The proof is in the Appendix.

8 Transfers among three or more agents

The two benchmark methods S^+ and S^-, and their affine combinations, are not vulnerable to bilateral partial transfers of jobs, but trilateral transfers can be a problem.

A simple example with $N = \{1, 2, 3, 4\}$ illustrates this important point. In the problem $x = (1, 1, 8, 3)$ coalition $T = \{1, 2, 3\}$ rearranges its three jobs as $x' = (2, 4, 4, 3)$. The actual wait of everyone in T is the same at x and at the reported x': in the latter, the slot $x'_1 = 2$ is used to complete jobs x_1 and

x_2, whereas the slots $x_2' = x_3' = 4$ are devoted to job x_3. We check that under both S^+ and S^-, the total tax on T decreases from x to x'. Equivalently, the tax on agent 4 increases. By Definition 2

$$\begin{aligned} \text{under } S^+ \text{ at } x \quad &: \quad t_4^+ = \frac{1}{2}; \text{ at } x' : (t_4^+)' = 2 \\ \text{under } S^- \text{ at } x \quad &: \quad t_4^- = 1; \text{ at } x' : (t_4^-)' = \frac{5}{2} \end{aligned}$$

Now *any* mechanism described in statement $i)$ of Theorem 2 is vulnerable to the same trilateral transfer: indeed $\alpha(x_N)$ and $\gamma(x_N)$ do not change from x to x'. Therefore we have proved

Corollary 4 to Theorem 2 *If $|N| \geq 4$, any efficient and continuous mechanism is vulnerable to job transfers involving three or more agents.*

For the sake of brevity, we do not give a formal definition of profitable transfers of jobs involving 3 or more agents. The definition is notationally cumbersome, and brings no additional intuition beyond that provided by the numerical example above. Notice that the shift from x to x' may be interpreted as the combination of merging jobs x_1, x_2 and splitting job x_3. This suggests that our first negative result, Theorem 1, is closely related to Corollary 4.

9 Concluding comment

The equality of waiting costs accross agents is an important simplifying assumption in our model. When we allow arbitrary linear waiting costs, the two solutions S^+, S^-, are easily extended (see Chun [2004b], who offers a characterization based on Consistency). But the interpretation of Merge-proofness becomes problematic: which waiting cost will the merged coalition adopt ? By merging with a null agent whose waiting cost is different, an agent can effectively misreport his own cost, hence the S^+ mechanism becomes vulnerable to merging, even if the server has full knowledge of individual waiting costs. On the other hand splitting maneuvers are unambiguous in this context, provided all aliases have the same waiting cost as the true agent; and transferring tactics are similarly well defined. It is easy to check that the extended S^+ and S^- solutions are still pairwise transfer-proof, and the latter is split-proof.

Whether or not a characterization result similar to Theorem 2 holds in this context is left for future research.

References

[1] Aczél, J. 1996. *Lectures on Functional Equations and Their Applications.* New York: Academic.

[2] Banker, R. 1981. Equity consideration in traditional full-cost allocation practices: an axiomatic perspective, *Joint Cost Allocations*, S. Moriarty, ed. Norman: University of Oklahoma.

[3] Chun, Y. 2000. Agreement, Separability, and other Axioms for Quasi-linear Social Choice Problems, *Social Choice and Welfare,* 17, 507-521.

[4] Chun, Y., 2004. A Note on Maniquet's Characterizations of the Shapley Value in Queuing Problems, mimeo.

[5] Chun, Y., 2004. Consistency and Monotonicity in Sequencing Problems, mimeo.

[6] Curiel, I., G. Pederzoli and S. Tijs, 1989. Sequencing Games, *European Journal of Operational Research*, 40, 344-351.

[7] Curiel, I., J. Potters, V. Rajendra Prasad, S. Tijs and B. Veltman, 1993. Cooperation in One Machine Scheduling, *Methods of Operations Research*, 38, 113-131.

[8] Curiel, Il, H. Hamers and F. Klijn, 2002. Sequencing Games: a Survey, in *Chapters in Game Theory in Honor of Stef Tijs*, P. Borm and H. Peters (eds.), pp. 27-50, Kluwer Academic Publishers, Boston.

[9] De Frutos, M.A. 1999. Coalitional Manipulations in a Bankruptcy Problem, *Review of Economic Design,* 4,255-272.

[10] Dolan, R., 1978. Incentive Mechanisms for Priority Queueing Problems, *The Bell Journal of Economics* 9, 421-436.

[11] Douceur, J. 2002 The Sybil Attack, *Proceedings of the 1st International Workshop on Peer-to-Peer Systems* (IPTPS 2002), LNCS 2429, pp 251-260.

[12] Ermolov, A. 1995. Coalitional Manipulation in a Quasi-linear Economy, *Games and Economic Behavior,* 8, 349-363.

[13] Friedman E. and Moulin H., 1999. Three Methods to Share Joint Costs or Surplus, *Journal of Economic Theory, 87, 2, 275-312.*

[14] Hain, R. and M. Mitra, 2001. Simple Sequencing Problems with Interdependent Costs, mimeo, University of Bonn.

[15] Hamers, H., J. Suijs, S. Tijs, and P. Borm, 1996. The Split Core for Sequencing Games, *Games and Economic Behavior* 15, 165-176.

[16] Ju, B.G., 2003. Manipulations via Merging and Splitting in claim problems, *Review of Economic Design* 8, 205-215.

[17] Ju, B.G. and E. Miyagawa, 2003. Coalitional Manipulations and Generalized Proportional Rules, mimeo, Columbia University.

[18] Katta, A. and J. Sethuraman, 2004. A note on Cooperation in Queues, mimeo.

[19] Kittsteiner, T. and B. Moldovanu, 2003. Auction-Based Queue Disciplines, mimeo, University of Bonn.

[20] Kittsteiner, T. and B. Moldovanu, 2003. Priority Auctions and Queue Disciplines that Depend on Processing Time, mimeo, University of Bonn.

[21] Klijn, F. and E. Sánchez, 2002. Sequencing Games Without a Completely Specific Initial Order, *Reports in Statistics and Operations Research*, Report 02-04, 1-17.

[22] Maniquet, F., 2003. A Characterization of the Shapley Value in Queueing Problems, *Journal of Economic Theory* 109/1, 90-103.

[23] Mendelson, H. and S. Whang, 1990. Optimal Incentive Compatible Priority Pricing for the M/M/1 Queue, *Operations Research* 38, 870-883.

[24] Mitra, M. and A. Sen, 1998. Dominant Strategy Implementation of First-Best Public Decisions, Indian Statistical Institute Delhi Centre, Manuscript No. 98-02.

[25] Mitra, M., 2001. Mechanism Design in Queueing Problems, *Economic Theory* 17, 277-305.

[26] Mitra, M., 2002. Achieving the First Best in Sequencing Problems, *Review of Economic Design* 7, 75-91.

[27] Moulin, H., 1985. Egalitarianism and Utilitarianism in Quasi-linear Bargaining, *Econometrica,* 53, 1, 49-67.

[28] Moulin, H., 1987. Equal or Proportional Division of a Surplus, and Other Methods, *International Journal of Game Theory*, 16, 3, 161–186.

[29] Moulin, H., 2004. Split-proof Probabilistic Scheduling, mimeo.

[30] Smith, W., 1956. Various Optimizers for Single-Stage Production, *Naval Res. Logistics Quarterly* 3, 59-66.

[31] Sprumont, Y., 2003. Aumann-Shapley pricing: a reconsideration of the discrete case, mimeo.

[32] Suijs, J., 1996. On Incentive Compatibility and Budget Balancedness in Public Decision Making, *Economic Design* 2, 193-209.

10 Appendix

10.1 Theorem 1

We fix $\mathcal{N}, |\mathcal{N}| \geq 4$ and a mechanism μ satisfying MPF and SPF, as well as either CONT and ETE, and we derive a contradiction. Recall that MPF implies efficiency.

Step 1 *A limited symmetry property*
Fix $N, |N| \geq 2$, and two agents $1, 2 \in N$. Fix any $x \in \mathbb{R}^N$ such that $x_1 = x_2 = a > 0$. We write x_{-1}, x_{-2} for its projection on $N \backslash 1$ and $N \backslash 2$ respectively, and define $z^1, z^2 \in \mathbb{R}^N$ by $z^1_2 = 0, z^1_{-2} = x_{-2}$, and $z^2_1 = 0, z^2_{-1} = x_{-1}$.We claim

$$y_{12}(N, z^1) = y_{12}(N, z^2) = y_1(N \backslash 2, x_{-2}) = y_2(N \backslash 1, x_{-1})$$

In the merging of 1 and 2 in z^1 to agent 1 in x_{-2}, merge-proofness (4) gives

$$y_{12}(z^1) \leq a + x_{P(1,\sigma^*)} + t^*_1(x_{-2}) = y_1(x_{-2})$$

because $k(\{1, 2\}, z^1) = 1$. In the split of 1 in x_{-2} to agents $1, 2$ in z^1, split-proofness implies

$$y_1(x_{-2}) \leq a + x_{P(1,\sigma_*)} + (t_*)_{12}(z^1) = y_{12}(z^1)$$

because 1 is the only agent with a positive job in $S = \{1, 2\}$. Thus we get $y_{12}(z^1) = y_1(x_{-2})$. Consider similarly the merging of 1,2 in z^1 to agent 2 in x_{-1}, and the split of 2 in x_{-1} to agents 1,2 in z^1 : we get $y_{12}(z^1) = y_2(x_{-1})$. Exchanging the roles of 1,2 gives the remaining equalities in the claim.

Step 2 The case of two agents problems

Fix a vector $(a, b) \in \mathbb{R}^2_+$ s.t. $0 < a \leq b$, and an arbitrary triple $N = \{1, 2, 3\}$ in $\mathcal{N}$. From Step 1 applied to $x = (a, a, b)$ we get

$$y_1(\{13\}, (a, b)) = y_2(\{23\}, (a, b))$$

Set $y_i(\{i, j\}, (a, b)) = u_i(ij)$ and $y_j(\{i, j\}, (a, b)) = v_j(ij)$. We have proven $u_i(ij) = u_k(kj)$ for i, j, k all distinct. Applying similarly Step 1 to $x = (a, b, b)$ we get $v_j(ij) = v_k(ik)$. Efficiency implies $u_i(ij) + v_j(ij) = 2a + b$, therefore $v_j(kj) = v_j(ij)$ and $u_i(ij) = u_i(ik)$. We can now set $u_i = u_i(ij)$ and $v_i = v_i(ji)$ for all $j \in N \backslash i$. Efficiency shows that $u_i + v_j$ does not depend on the pair (i, j) in $\mathcal{N}$, therefore u_i and v_i are both independent of $i \in \mathcal{N}$. We define a function $f(a, b)$ as follows

$$y_i(\{i, j\}, (a, b)) = a + f(a, b);\ y_j(\{i, j\}, (a, b)) = a + b - f(a, b)$$

keeping in mind that the pair (i, j) is arbitrary and $a \leq b$.

Step 3

We now compute explicitely the vector of transfers for a three-person problem $N = \{1, 2, 3\}$ and $x = (a, b, c)$ with $0 < a \leq b$ and $a + b \leq c$. This shows in particular that it does not depend on the choice of a triple in $\mathcal{N}$.

Consider the merging of 2,3 in x to 2 in $x^* = (a, b+c)$. As the total physical wait of agents $2, 3$ is the same before and after merging, MPF implies

$$t_{23}(N, x) \leq t_2(\{1, 2\}, x^*) = -f(a, b + c)$$

When splitting agent 2 in x^* to $2, 3$ in x, the physical wait of agent 2 is similarly constant, thus SPF implies the opposite inequality and we get $t_{23}(x) = -f(a, b + c)$.

Next consider the merging of 1,2 in x to 1 in $x_* = (a + b, c)$, and the symmetrical split of 1 in x_* to 1,2 in x. Our choice of a, b, c guarantees that the actual wait of 1,2 is constant in the merging, and that of 1 is constant in the split. Therefore $t_{12}(x) = f(a+b, c)$. Because $t_{123} = 0$ the vector $t(x)$ is now computed explicitely. It is convenient to use instead of f the function g defined by $f(\alpha, \beta) = g(\alpha, \alpha + \beta)$ for all $\alpha, \beta \geq 0$. We have:

$$t(N, x) = (g(a, d), g(a+b, d) - g(a, d), -g(a+b, d)) \tag{12}$$

for all a, b, d such that $0 < a \leq b$, and $2(a+b) \leq d$.

Next we invoke ETE or CONT at such a triple (a, b, d) where $a = b$. ETE implies $t_1 = t_2 + a$. On the other hand CONT implies that f is continuous in both variables, and so is g. For a small positive ε, the net waiting cost of agent 1 at $(a-\varepsilon, a, d)$ is $g(a-\varepsilon, d)+a-\varepsilon$, and it is $g(2a+\varepsilon, d)-g(a, d)+2a+\varepsilon$ at $(a+\varepsilon, a, d)$. By continuity, $t_1 = t_2 + a$ follows, namely

$$g(2a, d) = 2g(a, d) - a \text{ for all } a, 0 < a \leq \frac{d}{2} \tag{13}$$

The last step of the proof extends the above argument to four agents problems like $x = (a, b, c, d-(a+b+c))$ with $0 < a \leq b \leq c$ and $2(a+b+c) \leq d$. Looking to the merging of 3,4 and its reverse split, we deduce $t_{34}(x) = -g(a+b, d)$ from (12). From the merging of $2, 3, 4$ and the reverse split, we get $t_{234}(x) = -g(a, d)$, and finally from the merging of 1,2,3 and its reverse split we have $t_{123}(x) = g(a+b+c, d)$. Gathering our results

$$t(x) = (g(a, d), g(a+b, d)-g(a, d), g(a+b+c, d)-g(a+b, d), -g(a+b+c, d))$$

At a profile x where $b = c$, ETE or the same continuity argument as above gives

$$g(a+b, d) - g(a, d) = g(a+2b, d) - g(a+b, d) + b \tag{14}$$

We derive finally a contradiction between (14) and (13). Taking $a = b$ in (14), and omitting d for simplicity, we get

$$g(3a) = 2g(2a) - g(a) - a = 3g(a) - 3a$$

Taking $b = 2a$ gives similarly $g(5a) = 5g(a) - 8a$. Finally taking $a = 2x, b = 3x$ in (14) again gives

$$g(8x) = 2g(5x) - g(2x) - 3x = 10g(x) - g(2x) - 19x$$

On the other hand (13) is $g(2x) = 2g(x) - x$, and implies $g(8x) = 8g(x) - 12x$. A contradiction follows.

10.2 Theorem 2

10.2.1 Proof of Statement i

Consider the method associated with the functions α and γ. As a spread or a shrink leaves the sum x_N, and therefore $\gamma(x_N)$, unchanged, we can simply ignore γ while checking PTP. Recall that y^+ and y^- are separable with associated functions θ^+ and θ^-. Thus $y = \alpha y^+ + (1-\alpha)y^-$ can be written

$$y_i(N,x) = x_i + \sum_{j \in N \setminus i} \theta(x_i, x_j; x_N)$$

where $\theta(a,b;z) = \alpha(z)\theta^+(a,b) + (1-\alpha(z))\theta^-(a,b) = \frac{1}{2}(a \wedge b) + \frac{1}{2}(\alpha(z) - 1)(a-b)$.

For fixed b and z, the function $a \longrightarrow \theta(a,b;z)$ is linear before b and linear after b, and its slope drops by $\frac{1}{2}$ at b. In particular, this function is concave. Thus all we need to show is that any mechanism coming from the method y meets PTP.

Consider first S, x, x' and ε as in the premises of (10). Assume without loss of generality $x_1 < x_2$ and that agent $2 \in S$ transfers ε of his job to $1 \in S$. As $x_N = x'_N$, we omit x_N in $\theta(x_i, x_j; x_N)$ and compute the total net cost of S before and after the shrink:

$$y_S(x) = v(S,x) + \sum_{N \setminus S} [\theta(x_1, x_j) + \theta(x_2, x_j)]$$

$$y_S(x') = v(S,x') + \sum_{N \setminus S} [\theta(x'_1, x_j) + \theta(x'_2, x_j)]$$

$$v(S,x') - v(S,x) = (2x'_1 + x'_2) - (2x_1 + x_2) + \sum_{k \in S \setminus \{1,2\}} p_k$$

where the term $p_k = x'_1 \wedge x_k + x'_2 \wedge x_k - x_1 \wedge x_k - x_2 \wedge x_k$ is nonnegative because $a \to a \wedge x_k$ is concave. Therefore $v(S,x') - v(S,x) \geq \varepsilon$. The same concavity argument shows $\theta(x'_1, x_j) + \theta(x'_2, x_j) \geq \theta(x_1, x_j) + \theta(x_2, x_j)$, and the proof of (10) is complete.

Next we consider a spread, namely S, x, x' and ε as in the premises of (11) with $x_1 \leq x_2$ and $1 \in S$ transferring ε of her job to $2 \in S$. With the same notation p_k as above, we get: $v(S,x') - v(S,x) = -\varepsilon + \sum_{S \setminus \{1,2\}} p_k$. Setting $q_j = \theta(x'_1, x_j) + \theta(x'_2, x_j) - \theta(x_1, x_j) - \theta(x_2, x_j)$, we now have

$$y_S(x') + \varepsilon - y_S(x) = \sum_{S\backslash\{1,2\}} p_k + \sum_{N\backslash S} q_j \tag{15}$$

where the concavity argument shows this time $p_k \leq 0$ and $q_j \leq 0$. Check first that for any agent $i \notin \Delta(\sigma'; 1, 2)$, we have $p_i = 0$ if $i \in S\backslash\{1,2\}$ and $q_i = 0$ if $i \in N\backslash S$. This is clear because the functions $a \to a \wedge x_i$ and $a \to \theta(a, x_i)$ are linear on $[0, x_i]$ and on $[x_i, +\infty[$. Next we pick $i \in \Delta(\sigma'; 1, 2)$ and suppose first $i \in S\backslash\{1,2\}$. We have

$$x'_1 \leq x_i \leq x'_2 \Longrightarrow p_i = x'_1 + x_i - x_1 \wedge x_i - x_2 \wedge x_i \Rightarrow p_i \geq -x_i$$

Finally consider $i \in \Delta(\sigma'; 1, 2) \cap N\backslash S$. If we show $q_i + x_i \geq 0$, the desired inequality (11) will follow from (15). Recall that on the interval $[x'_1, x'_2]$, the function $a \to \theta(a, x_i)$ has 2 linear pieces connecting at x_i and such that the slope drops by $\frac{1}{2}$ at x_i. Therefore

$$q_i = (\theta(x'_2, x_i) - \theta(x_2, x_i)) - (\theta(x_1, x_i) - \theta(x'_1, x_i)) \geq -\frac{1}{2}(x_1 - x'_1)$$

The inequality $q_i + x_i \geq 0$ follows if $x_i \geq x_1$. If $x'_1 \leq x_i \leq x_1$, compute $q_i = \frac{x'_1 - x_i}{2}$, ensuring $q_i + x_i \geq 0$. This concludes the proof of statement i.

10.2.2 Proof of Statement ii

We fix N, and an efficient mechanism μ meeting CONT and PTP.

Step 1

For all nonempty and proper subset S of N, we write $H(S) = \{x \in \mathbb{R}^N_+ | x_i < x_j \text{ for } i \in S, j \in N\backslash S\}$. We prove the existence of a function $g^S(a, b)$ such that

$$\begin{aligned} g^S(a,b) \text{ is defined for } a, b &\geq 0 \text{ such that } \frac{a}{|S|} < \frac{b}{|N|} \\ g^S(x_S, x_N) &= t_S(x) \text{ for all } x \in H(S) \end{aligned} \tag{16}$$

where $t(x)$ is the monetary transfer selected by μ.

For any $x \in H(S)$, efficiency of μ implies that $\sigma(x)$ ranks S ahead of $N\backslash S$, therefore $y_S(x) = v(S, x) + t_S(x)$. Given $x \in H(S)$, we define the vector x^*

by $x_i^* = \frac{x_S}{|S|}$ if $i \in S, x_i^* = x_i$ if $i \in N \backslash S$. Note that x^* is also in $H(x)$. Our first step toward proving (16) is to show $t_S(x) = t_S(x^*)$.

We call two agents i, j adjacent at x if $\Delta(\sigma(x); i, j) = \varnothing$. Given x and $i, j \in S$, adjacent at x, consider x' obtained from x by averaging x_i and $x_j : x_i' = x_j' = \frac{1}{2}(x_i + x_j), x_k' = x_k$ otherwise. Thus x' is a shrink of x, and x a spread of x', and PTP implies $y_S(x) = y_S(x') - \varepsilon$, where $\varepsilon = \frac{1}{2}|x_i - x_j|$.

Note that x' is in $H(S)$ as well, and that $v(S, x') = v(S, x) - \varepsilon$, because $]x_i, x_j[$ contains no $x_k, k \neq i, j$.Now $y_S(x) = v(S, x) + t_S(x)$ and the similar equality for x' imply $t_S(x) = t_S(x')$.

For any $x \in H(S)$ such that $x \neq x^*$, we can find two agents $i, j \in S$, adjacent at x, and average x_i and x_j without changing $t_S(x)$. Thus we construct a sequence $x^\circ = x, x^1, x^2, ...$, by averaging at each step some pair x_i, x_j where i, j are adjacent at x. This sequence either stops at x^* or converges to x^*. By construction

$$y_S(x^p) = v(S, x^p) + t_S(x) \text{ for } p = 0, 1, 2, ...$$

By continuity of y_S and of $v(S, .)$, we deduce $y_S(x^*) = v(S, x^*) + t_S(x)$; because $x^* \in H(x)$, this gives $t_S(x) = t_S(x^*)$ as claimed.

A symmetrical construction, starting from any $x \in H(S)$, and successively averaging x_i, x_j for some $i, j \in N \backslash S$ adjacent at x, delivers $t_{N \backslash S}(x) = t_{N \backslash S}(x_*)$ where $(x_*)_i = \frac{x_{N \backslash S}}{|N \backslash S|}$ if $i \in N \backslash S$ and $(x_*)_i = x_i$ if $i \in S$. Combining this with $t_S(x) = t_S(x^*)$, and $t_S + t_{N \backslash S} = 0$, we see that $t_S(x)$ only depends upon x_S and $x_{N \backslash S}$, and can be written as in (16) for some function g^S. Finally $x \in H(S)$ implies $\frac{x_S}{|S|} < \frac{x_N}{|N|}$ and the proof of Step 1 is complete.

In the next step we use the following consequence of (16). If at problem (N, x) we have $x_j < x_i < x_k$ for all $j \in S$ and all $k \in N \diagdown S \cup \{i\}$, then $t_i(x) = g^{S \cup \{i\}}(x_{S \cup \{i\}}, x_N) - g^S(x_S, x_N)$. This holds even if $S = \varnothing$, by setting $g^\varnothing = 0$, and also if $S \cup \{i\} = N$, by setting $g^N = 0$.

Step 2

We observe first that each function g^S is continuous on its domain. For each $a, b \geq 0$ such that $\frac{a}{|S|} < \frac{b}{|N|}$, we define $x(a, b) = z$ by $z_i = \frac{a}{|S|}$ for $i \in S$, and $z_i = \frac{b-a}{|N \backslash S|}$ for $i \in N \backslash S$. By Step 1

$$g^S(a, b) = t_S(x(a, b)) = y_S(x(a, b)) - v(S, x(a, b))$$

and the claim follows by CONT. Next we apply continuity again at those profiles where two coordinates are equal, and derive a functional equation

((17) below) linking the different functions g^S.

In the rest of the proof we use the simplified notation $S \cup \{i\} = S, i$, $\{i, j\} = i, j$, etc...

Fix S nonempty, and two agents $i, j \in N \backslash S$. Choose also any three a, b, c such that $0 \leq a < b < c$. We construct x and, for ε small enough, $x(\varepsilon)$ as follows:

$$x_k = a \text{ if } k \in S; x_i = x_j = b; x_k = c \text{ if } k \in N \diagdown S, i, j$$
$$x(\varepsilon) = x + \varepsilon(e^i - e^j) \text{ where } e^i \text{ is the } i-th \text{ unit vector in } \mathbb{R}^N.$$

For ε small enough and positive, any efficient ordering of N ranks S before j, j before i, and i before the rest. For ε small and negative, the order is $S \prec i \prec j \prec N \backslash S, i, j$. From Step 1, we have

$$y_i(x(\varepsilon)) = x_{S,i,j} + t_i(x(\varepsilon)) = x_{S,i,j} + g^{S,i,j}(x_{S,i,j}, x_N) - g^{S,j}(x_{S,j} - \varepsilon, x_N) \text{ for } \varepsilon > 0$$

$$y_i(x(\varepsilon)) = x_{S,i} - \varepsilon + g^{S,i}(x_{S,i} - \varepsilon, x_N) - g^S(x_S, x_N) \text{ for } \varepsilon < 0$$

By continuity of y_i and of g^T, for all T, we deduce

$$b + g^{S,i,j}(sa + 2b, d) - g^{S,j}(sa + b, d) = g^{S,i}(sa + b, d) - g^S(sa, d) \qquad (17)$$

where $s = |S|, n = |N|$ and $d = sa + 2b + (n - s - 2)c$. Our choice of c is only limited by $0 \leq a < b < c$. Thus if $S, i, j \neq N$ equation (17) holds for all a, b, d such that $0 \leq a < b$ and $d > sa + (n - s)b$. In the case $S, i, j = N$, (17) holds by our convention $g^N \equiv 0$ and in that case we have $0 \leq a < b$ and $d = sa + 2b$.

Finally the continuity argument applies also to the case $S = \varnothing, a = 0$. Thus (17) holds in this case as well for $0 < b < d$ (recall our convention $g^{\varnothing} = 0$).

Step 3

We derive a first consequence of (17)

$$g^S(sb, d) = \sum_S g^i(b, d) - \frac{s(s-1)}{2} b \text{ for all } \varnothing \neq S \neq N, \text{ and all } 0 < b < \frac{d}{n} \qquad (18)$$

Equation (17) for $S = \varnothing, a = 0$, gives (18) for $S = i, j$. Apply (17) next to $S = k$ and $a < b, d > nb$.

$$g^{i,j,k}(a + 2b, d) = (g^{k,i} + g^{k,j})(a + b, d) - g^k(a, d) - b \qquad (19)$$

Fix d, let a converge to b, and use (18) for $S = k, i$ and $S = k, j$: we obtain (18) for $S = k, i, j$. An easy induction argument, omitted for brevity, concludes Step 3.

Step 4

We prove that each function $g^i(a, d)$ is affine in a, and its slope is independent of $i \in N$. The assumption $|N| \geq 4$ plays a key role in this step, and in this one only.

Develop (19) using (18) successively for $S = i, j, k, S = k, i$ and $S = k, j$. We get

$$(g^i + g^j + g^k)(\frac{a + 2b}{3}, d) = 2g^k(\frac{a + b}{2}, d) + (g^i + g^j)(\frac{a + b}{2}, d) - g^k(a, d)$$

for $0 < a < b$ and $\frac{a+2b}{3} < \frac{d}{n}$. As the choice of i, j, k in N is arbitrary, the term $g^k(\frac{a+b}{2}, d) - g^k(a, d)$ is independent of $k \in N$. Set it equal to $h(a, b, d)$ so the equation above becomes

$$\sum_{\omega=i,j,k} g^\omega(\frac{a + 2b}{3}, d) - g^\omega(\frac{a + b}{2}, d) = h(a, b, d)$$

As $|N| \geq 4$, and i, j, k are arbitrary, this implies for all i, $g^i(\frac{a+2b}{3}, d) - g^i(\frac{a+b}{2}, d) = \frac{1}{3}h(a, b, d)$. Thus, for fixed d, every function g^i meets the equation

$$g(\frac{a + b}{2}, d) = \frac{1}{4}g(a, d) + \frac{3}{4}g(\frac{a + 2b}{3}, d)$$

Changing variables to $a' = \frac{a+2b}{3}$ we get

$$g(\frac{1}{4}a + \frac{3}{4}a') = \frac{1}{4}g(a) + \frac{3}{4}g(a') \text{ for all } 0 \leq a < a' < \frac{d}{n}$$

where we omit d for simplicity. This is a simple variant of the classic Cauchy equation (see Aczel [1970]). As $g(., d)$ is continuous on the interval $[0, d[$, it must be affine, namely $g(a, d) = \lambda(d)a + \beta(d)$. Back to the functions g^i, recall that $g^i(\frac{a+b}{2}) - g^i(a)$ is independent of i: thus the slope $\lambda(d)$ is the same for all i and we conclude

$$g^i(a,d) = \lambda(d)a + \beta_i(d) \text{ for all } 0 < a < \frac{d}{n} \tag{20}$$

Step 5 *End of proof*

As discussed at the end of Step 2, we can apply (17) to $S = N \diagdown i, j, a, b$ such that $0 \leq a < b$ and $d = (n-2)a + 2b$. We obtain one more equation connecting $\lambda, \beta_i, i \in N$:

$$b = (g^{N \diagdown i} + g^{N \diagdown j})((n-2)a + b, d) - g^{N \diagdown i,j}((n-2)a, d) \tag{21}$$

Now (18) and (20) give

$$\begin{aligned} g^{N \diagdown i}((n-2)a + b, d) &= (\lambda(d) - \frac{n-2}{2}) \cdot ((n-2)a + b) + \beta_{N \diagdown i}(d), \\ g^{N \diagdown i,j}((n-2)a, d) &= (\lambda(d) - \frac{n-3}{2})(n-2)a + \beta_{N \diagdown i,j}(d) \end{aligned}$$

and we omit the similar formula for $g^{N \diagdown j}$. Upon substituting in (21):

$$b = \beta_N(d) + (\lambda(d) - \frac{n-2}{2})((n-2)a + 2b) - \frac{1}{2}(n-2)a \Longleftrightarrow \beta_N(d) = \frac{n-1}{2}d - \lambda(d)d$$

Now we set

$$\alpha = \frac{2\lambda + 1}{n} \Longleftrightarrow \lambda = \frac{n\alpha - 1}{2}; \text{and } \beta_i = \frac{1}{n}(\frac{n-1}{2} - \lambda)d + \gamma_i$$

where α and $\gamma_i, i \in N$, depend on d. From the continuity of g^i in a, d follows that of β_i and of λ in d, hence of α and γ_i in d. Moreover $\gamma_N \equiv 0$ by construction. We compute now, with the help of (18), g^i and g^S in terms of α and γ_i :

$$g^i(a,d) = \lambda a + \beta_i = \frac{n\alpha - 1}{2}a + \frac{1-\alpha}{2}d + \gamma_i;$$

$$g^S(a,d) = \frac{n\alpha - s}{2}a + \frac{1-\alpha}{2}sd + \gamma_S = \alpha\frac{(n-s)}{2}a + (1-\alpha)\frac{s(d-a)}{2} + \gamma_S$$

For our two basic methods y^+ and y^-, it is easy from Definition 2 to compute $t_S^+(x), t_S^-(x)$ whenever $x \in H(S)$:

$$t_S^+(x) = \frac{1}{2}(n-s)\cdot x_S; \quad t_S^-(x) = \frac{1}{2}s\cdot x_{N\backslash S}$$

Compare with the sum of transfers to S in our mechanism μ, namely $t_S(x) = g^S(x_S, x_N)$:

$$t_S(x) = \alpha(x_N)\cdot t_S^+(x) + (1-\alpha(x_N))\cdot t_S^-(x) + \gamma_S(x_N) \text{ for all } x \in H(S)$$

Recall, for any efficient method, any S and any $x \in H(S)$, the equation $y_S(x) = v(S,x) + t_S(x)$. We have just proven that the method y associated with μ, and the method $\widetilde{y} = \alpha y^+ + (1-\alpha)y^- + \gamma$ have $\widetilde{y}_S(x) = y_S(x)$ for all S and $x \in H(S)$. Now if all coordinates of x are different, this forces $y(x) = \widetilde{y}(x)$. By continuity the equality holds everywhere on $\mathbb{R}_+^N$. This concludes the proof of Theorem 2.

10.3 Corollaries of Theorem 2

10.3.1 Corollary 1

Statement i is obvious as y^+, y^- treat equals equally. For statement ii, the "if" part is obvious. To prove "only if," consider $x = d\cdot e^i$, where e^i is, as before, the i-th coordinate vector. Compute

$$y^+(x) = de^i; y^-(x) = \frac{d}{2}(1,..,1) - \frac{(n-1)}{2}de^i$$

Scale Invariance implies $y(x) = dy(e^i)$. Taking the j-th coordinate of this equation for $j \neq i$, gives

$$(1-\alpha(d))\frac{d}{2} + \gamma_j(d) = \frac{1-\alpha(1)}{2} + d\gamma_j(1)$$

As j varies in N and $\gamma_N \equiv 0$, this implies first $\alpha(d) = \alpha(1)$, then $\gamma_j(d) = d\gamma_j(1)$ for all j, as claimed.

10.3.2 Corollary 2

Clearly the component $\gamma(x_N)$ in $y(x)$ plays no role in the properties of merge-proofness and splitproofness, so we can assume $\gamma \equiv 0$ without loss of generality. Observe that the method $\alpha\cdot y^+ + (1-\alpha)\cdot y^-$ behaves essentially like a separable method with respect to the function

$$\theta^{\alpha}(a,b,d) = \frac{1}{2}(a \wedge b) - \frac{(1-\alpha(d))}{2}(a-b)$$

That is to say, the net cost $y_i(N,x)$ is computed as $y_i(N,x) = x_i + \sum_{N \backslash i} \theta^{\alpha}(x_i, x_j, x_N)$ for all N, i and x. We can then mimick the proof of Proposition 2: any mechanism with method $\alpha(x_N) \cdot y^+ + (1-\alpha(x_N)) \cdot y^-$ is mergeproof if and only if, for any fixed d, the function $\theta^{\alpha}(\cdot,\cdot,d)$ meets the system (6); any such mechanism is split-proof if and only if $\theta^{\alpha}(\cdot,\cdot,d)$ meets system (7).

One consequence of MPF is that $a \longrightarrow \theta(a,b,d)$ is superadditive on $[0,b]$. In particular

$$2\theta(\frac{b}{2}) \le \theta(b,b) \Longleftrightarrow \frac{b}{2} + (1-\alpha)\frac{b}{2} \le \frac{b}{2} \Longrightarrow \alpha \ge 1$$

Conversely, θ^{α} meets (6) whenever $\alpha \ge 1$. Indeed $a \longrightarrow \theta^{\alpha}(a,b,d)$ has slope $\frac{\alpha}{2} \ge \frac{1}{2}$on $[0,b]$ with $\theta(b) = \frac{b}{2}$, therefore $\theta(0) \le 0$. On $[0,b]$ we have $\theta(a_1 + a_2) - \theta(a_2) = \theta(a_1) - \theta(0)$, and the top inequality in (6) follows. The bottom one is equally easy.

A consequence of SPF is $\theta(2b,b,d) + b \le 2\theta(b,b,d)$, from the top inequality in (7). This amounts to $\theta(2b,b,d) \le 0 \Longrightarrow \frac{1}{2}b - \frac{(1-\alpha)}{2}b \le 0 \Longleftrightarrow \alpha \le 0$. Checking that, conversely, θ^{α} meets (7) whenever $\alpha \le 0$ is routine and omitted.

10.3.3 Corollary 3

Choose α, γ as in the statement of Theorem 2. When does the corresponding method meet ZCNJ? For all x, all $i \in N, x_i = 0$ implies $y_i^+(x) = 0$ and $y_i^-(x) = \frac{1}{2}x_N$. Therefore ZCNJ implies $\frac{1}{2}(1-\alpha(d))d + \gamma_i(d) = 0$ for all i and all $d \ge 0$. From this, $\gamma \equiv 0$ and $\alpha(d) = 1$ for all d follow at once.

Suppose next that the method associated with α, γ meets SAB. Apply this property first for x, i such that $x_i = 0$. We get $\frac{1}{2}(1-\alpha)d + \gamma_i \ge 0$. Summing over i gives $\alpha(d) \le 1$ for all d. Apply next SAB to $x = de^i$ and to agent i

$$\begin{aligned} d &\le \alpha y_i^+(x) + (1-\alpha)y_i^-(x) = \alpha d + (1-\alpha)\frac{3-n}{2}d + \gamma_i \qquad (22) \\ &\Longleftrightarrow \frac{n-1}{2}(\alpha-1)d + \gamma_i \ge 0 \end{aligned}$$

Summing up over i yields $\alpha(d) \geq 1$. Thus $\alpha \equiv 1$ and the inequality above gives $\gamma \equiv 0$ as well.

10.3.4 Remark 3

The pessimistic stand alone bound applied to $x = de^i$ gives the inequality opposed to (22) above, hence $\alpha(d) \leq 1$ for all d. Mergeproofness, on the other hand, amounts to $\alpha(d) \geq 1$ for all d.

Next apply the unanimity bound to any x, i such that $x_i = 0$. We get $\frac{1}{2}(1-\alpha)d+\gamma_i \leq 0$, hence $\alpha(d) \geq 1$ by summing over i. Thus the combination of the bounds $y_i \leq x_N$ and $y_i \leq \frac{n+1}{2}x_i$ captures, again, the method S^+.

.

.