

Fair Allocation of Multiple Resources Using a Non-monetary Allocation Mechanism

Patrick Poullie and Burkhard Stiller

University of Zürich, Department of Informatics (IFI), Communication Systems
Group (CSG), Binzmühlestrasse 14, CH-8050 Zürich, Switzerland
`{poullie, stiller}@ifi.uzh.ch`

Abstract. The fair allocation of scarce resources is relevant to a wide field of applications. For example, cloud resources, such as CPU, RAM, disk space, and bandwidth, have to be shared. This paper presents a mechanism to find fair allocations of multiple divisible resources, which, contrary to other mechanisms, is applicable to but not limited to the example above. Wide applicability of the mechanism is achieved by designing it (1) to scale with the number of consumers and resources, (2) to allow for arbitrary preference functions of consumers, and (3) to not rely on monetary compensation. The mechanism uses a mathematical definition of greediness to balance resources consumers receive and thereby to compute a fair allocation.

Keywords: Resource Allocation, Fairness, Multiple Resources.

1 Problem Description and Related Work

The fair allocation of scarce resources is relevant to a wide field of applications. The fair sharing of bandwidth is a classical network resource allocation problem. However, shared systems also bear allocation problems where more than one resource has to be divided, e.g., the cloud computing paradigm allows different stakeholders to share different resources like CPU, RAM, disk space, and bandwidth. When demands for resources are different, it is necessary to allocate consumers different bundles in order to maximize satisfaction. In particular, allocating resources independently of each other (based on some fairness criteria, e.g., proportional fairness) may yield overall allocations that are inefficient. Finding an allocation that is fair and efficient is non trivial.

Formally an allocation problem can be defined as follows. Let $A = \{a_1, a_2, \dots, a_m\}$ be a set of consumers and $R = \{r_1, r_2, \dots, r_n\}$ be a set of resources. When resources are divisible function $q: R \rightarrow \mathbb{R}$ maps each resource to the amount that is available of it. Any partition of R into m bundles $\{b_1, b_2, \dots, b_m\}$ is called an allocation, whereas consumer a_i receives bundle b_i . Consumer a_i 's appreciation for a bundle is given by a_i 's preference function p_i that maps every bundle to a real number expressing the consumers valuation for the bundle, e.g., a_i prefers b_2 to b_1 , if $p_i(b_1) < p_i(b_2)$. The problem is to find an allocation with certain characteristics, where the two most prominent characteristics are fairness and

efficiency. In particular, an allocation has the characteristic that it is *fair*, if no consumer prefers another consumer's bundle to his bundle [1], which can be expressed mathematically as $\forall 1 \leq i, j \leq m: p_i(b_i) \geq p_i(b_j)$. Further, an allocation is efficient, if it is pareto-efficient, *i.e.*, no allocation exists that makes at least one consumer better off without making any other worse off [1].

An allocation mechanism is an algorithm to find an allocation for an allocation problem. Many allocation mechanisms are presented throughout literature and shown to produce fair or efficient allocations [1, 2]. Based on a requirements analysis the following three requirements for general applicability of allocation mechanisms were identified.

- (1) **Scalability** – For many applications, it is important that an allocation mechanism is computationally feasible despite a large number of consumers and resources. This is not the case for all allocation mechanisms, for example, [3] can ensure envy-freeness and pareto-efficiency but does not scale. Demanding scalability with the number of consumers implies that this number is not bounded. For example, [4, 5] are constrained to two consumers. Demanding scalability with the number of resources implies that this number is also not bounded. Scheduling algorithms typically consider the allocation of only one resource (or at least interchangeable resources). Hence, they cannot be applied, when two or more non-interchangeable resources, with inter-depending utilities, have to be allocated.
- (2) **Arbitrary preference functions** – It is desirable that allocation mechanisms need to make as little assumptions on consumers' preference functions as possible. For example, [1, p. 271 et seq.] and [6] assume that the valuation of a bundle does not depend on the ratio of resources in it. This assumption appears too simplistic for many use cases.
- (3) **No monetary compensation** – Many allocation mechanisms [7–9] involve money. It is assumed that if a consumer does not get a large bundle of resources, but therefore can hold on to most of its money, the outcome is fair. However, monetary compensation for missing resources may be perceived as a consolidation price.

Since no allocation mechanism exists, which suffices all three criteria, this proposal here closes this gap by an allocation mechanism for multiple divisible resources.

2 Approach

The allocation mechanism is round-based. Each round every consumer demands a bundle that he wishes to receive (which depends on his preference function). However, consumers will only receive bundles when the process stops, *i.e.*, after the last round. Demands of the same round are input to a mathematical function g to assess the proportionality of every demand. In particular, g maps every demand to a real number, that expresses its proportionality, and therefore can be associated to the *greediness* of the consumer that made it. The number is

returned to the respective consumer who *should* take it into account when making the demand in the next round, *i.e.*, if a consumer receives a number greater zero (which implies that he made an exproportional demand) he should demand a smaller bundle in the next round. Therefore, the progression of demands is influenced by the feedback consumers receive, which is determined by g . The definition of g is, therefore, closely considered in Section 2.1. When the process stops, each consumer receives the bundle he demanded in the last round. In case the sum of demands exceeds the resource supply, bundles have to be shaped, which is done in proportion to the consumers' greediness (which is defined by g and the last demanded bundle), *i.e.*, the bundle that a greedy consumer receives will be trimmed stronger than the bundle a moderate consumer receives. This gives incentives to consumers, to take the feedback about their greediness into account, as they may suffer from strongly trimmed resource bundles.

2.1 Greediness

For an arbitrary round in the allocation mechanism, let $d_{i,j} \geq 0$ be the demand of consumer a_i for resource r_j . For any resource r_j , amount $fs(r_j) = q(r_j) \div m$ is the *equal-share of resource r_j* for any consumer, *i.e.*, the resource is divided equally among all consumers. Then $d_{i,j} - fs(r_j)$ is the amount of resource r_j that consumer a_i demands beyond his equal-share (if the difference is negative, a_i is willing to release some of his equal-share). If $d_{i,j} > fs(r_j)$, other consumers have to release some of their equal-share of r_j in order to cover for a_i 's additional demand. Therefore, the additional demand should be added to the greediness of a_i . If $d_{i,j} \leq fs(r_j)$ consumer a_i 's greediness should decrease. However, in this case, only to an extend that other consumers benefit from the release, which applies, when other consumers request r_j beyond their equal-share. This notion can be expressed as follows. Define $\alpha(r_j)$ as the sum of what consumers demand beyond their equal-share of r_j , *i.e.*, $\alpha(r_j) = \sum_{i=1}^m \max(0, d_{i,j} - fs(r_j))$, and $\beta(r_j)$ as the sum of what consumers release of r_j , *i.e.*, $\beta(r_j) = -1 \cdot \sum_{i=1}^m \min(0, d_{i,j} - fs(r_j))$. The *greediness of consumer a_i* is defined as $g(a_i) = \sum_{j=1}^n g(d_{i,j})$, where

$$g(d_{i,j}) := \begin{cases} d_{i,j} - fs(r_j) & \text{if } d_{i,j} - fs(r_j) \geq 0 \\ (d_{i,j} - fs(r_j)) \cdot \min(1, \frac{\alpha(r_j)}{\beta(r_j)}) & \text{else.} \end{cases}$$

2.2 Meeting the Three Criteria

Because this allocation mechanism only considers demands made by consumers but not directly their preference functions, no assumptions need to be made on the preference functions (req. (2)). Since consumers do not need to communicate with each other but must announce their demands to a central entity (a condition that is given when users compete for cloud resources) that calculates g , consumers can run on different machines. This allows to add machines if the number of consumers is large (and run consumers with complex preference functions on dedicated machines). Because the computation of g is linear in the

number of consumers and resources (*cf.* Section 2.1), the central entity scales and, therefore, req. (1) is met. No monetary compensation is needed (req. (3)) to ensure a fair and efficient outcome, because the mechanism streamlines the proportionality of bundles consumers receive based on their equal-share.

3 Final Remarks

An allocation mechanism for multiple divisible resources and an arbitrary number of consumers was presented. It can be applied to cloud resource allocation problems but is not limited to such scenarios. The mechanism scales and does neither need to make assumptions on the consumers' preference functions nor does it use monetary compensation. Due to the comprehensible definition of greediness it utilizes, it is expected to yield fair and efficient allocations. Since resulting allocations depend on consumer behavior, game-theoretic analysis and simulations will be deployed to formally evaluate (i) how fair and efficient resulting allocations are and (ii) if it is prone to strategic demand statements.

Acknowledgments. This work was supported partially by the SmartenIT and the FLAMINGO projects funded by the EU FP7 Program under Contract No. FP7-2012-ICT-317846 and No. FP7-2012-ICT-318488, respectively. The authors wish to thank T. Bocek and M. Waldburger for their enlightening discussions and input.

References

1. Brams, S.: Mathematics and Democracy. Princeton University Press (2008)
2. Moulin, H.: Fair Division and Collective Welfare. The MIT Press (2004)
3. Herreiner, D., Puppe, C.: A Simple Procedure for Finding Equitable Allocations of Indivisible Goods. *Social Choice and Welfare* 19(2), 415–430 (2002)
4. Brams, S., Fishburn, P.: Fair Division of Indivisible Items Between Two People with Identical Preferences: Envy-freeness, Pareto-optimality, and Equity. *Social Choice and Welfare* 17(2), 247–267 (2000)
5. Brams, S., Kilgour, M., Klamler, C.: The Undercut Procedure: An Algorithm for the Envy-free Division of Indivisible Items. MPRA Paper 12774 (2009), <http://ideas.repec.org/p/pra/mprapa/12774.html>
6. Schulman, L., Vazirani, V.: Allocation of Divisible Goods Under Lexicographic Preferences. Computer Research Repository (2012), <http://arxiv.org/abs/1206.4366>
7. Yang, S., Hajek, B.: VCG-Kelly Mechanisms for Allocation of Divisible Goods: Adapting VCG Mechanisms to One-Dimensional Signals. *IEEE Journal on Selected Areas in Communications* 25(6), 1237–1243 (2007)
8. Chorppath, A., Bhashyam, S., Sundaresan, R.: Almost Budget-balanced Mechanisms for Allocation of Divisible Resources. In: 47th Annual Allerton Conference on Communication, Control, and Computing, pp. 1120–1125 (2009)
9. Jain, R., Walrand, J.: An Efficient Nash-implementation Mechanism for Network Resource Allocation. *Automatica* 46(8), 1276–1283 (2010)