Welfare Engineering in Practice: On the Variety of Multiagent Resource Allocation Problems

Yann Chevaleyre[1], Ulle Endriss[2], Sylvia Estivie[1], and Nicolas Maudet[1]

[1] LAMSADE, Université Paris IX-Dauphine
Paris 75775 Cedex 16 (France)
Email: {chevaley,estivie,maudet}@lamsade.dauphine.fr
[2] Department of Computing, Imperial College London
180 Queen's Gate, London SW7 2AZ (UK)
Email: ue@doc.ic.ac.uk

Abstract. Many problems studied in the multiagent systems community can be considered instances of an abstract multiagent resource allocation problem. In this problem, which is now better understood theoretically, the goal is to satisfy a criterion of global optimality (formulated in terms of a suitable notion of social welfare), given that the agents sharing a set of resources follow a local rationality criterion reflecting their individual preferences. In this paper, we first show that this simple decentralised resource allocation framework allows us to model a wide variety of applications. These applications thereby benefit from all the theoretical results concerning the framework. We then draw up a list of criteria which can guide the application designer working within the framework and illustrate the relevance of our approach by discussing several applications in view of this list of design criteria.

1 Introduction

In this paper, we further develop the framework of *welfare engineering* [1], which addresses the design of suitable rationality criteria for autonomous software agents participating in negotiations over resources in view of different notions of social welfare, as well as the development of such notions of social welfare themselves. That is, as we shall explain, welfare engineering is complementary to both mechanism design and classical welfare economics.

Several companion papers have studied the theoretical properties of the welfare engineering framework and have, in particular, been concerned with identifying appropriate rationality criteria for a given choice of the notion of social welfare used to assess the quality of negotiation outcomes [2, 3, 1]. The second aspect of welfare engineering, namely the design of suitable notions of social welfare in view of the properties of the application domain, however, has received less attention so far. As we shall illustrate in this paper, the strength of our framework lies in the fact that it is flexible enough to cater for a surprisingly large variety of application domains. All theoretical results concerning this framework thus apply to these applications. The flip-side of the coin is, of course,

that in absence of precise guidelines we may soon encounter difficult design issues when trying to translate real applications into our abstract framework. In particular, it leaves the designer with a difficult choice to make when having to decide which social welfare measure is the most appropriate in the context of a given application. The main purpose of this paper is to present a number of criteria (derived from constraints attached to the application domains) that will help the designer in this task, and to illustrate their relevance to several real applications.

The remainder of this paper is organised as follows. Section 2 introduces the general methodology underlying the welfare engineering approach and discusses its relation to classical welfare economics and mechanism design. The basic multiagent resource allocation framework used in this paper is defined in Section 3, which also discusses how to account for additional concepts such as monetary side payments or agent roles. Section 4 introduces the problem of the "designer scope" and Section 5 discusses several concrete applications that can be modelled as multiagent resource allocation problems. A preliminary list of design criteria for applications based on the abstract multiagent resource allocation framework is given in Section 6. These criteria are then discussed in view of the example applications introduced before. Our conclusions are presented in Section 7.

2   Welfare Engineering

As mentioned in the introduction, welfare engineering is closely related to both welfare economics and to mechanism design.

An important issue addressed by *welfare economics* (and specifically by *welfarism*) is the question of how to measure the well-being of society with respect to the welfare of individuals. Technically, an answer to this question can be formalised by defining a *social welfare ordering*, *i.e.* a mapping from a set of individual preference relations or utility functions to a societal preference relation over alternative states of affairs [4]. In the classical literature, the question as to what social welfare ordering is the right one is mostly discussed from a philosophical or ethical point of view [5–7]. Different answers to this question will typically claim to be rather general in scope (because they are derived from general ethical principles, for instance) and they are, of course, understood to apply to human society. In contrast to this, welfare engineering is concerned with choosing (and possibly designing) tailor-made social welfare orderings that are appropriate for *specific* applications; and the focus is on societies of *artificial* agents. An example is the *elitist* social welfare ordering [1], which favours states in which the most successful agent enjoys a very high utility. This would be considered inappropriate for assessing the welfare of human society, but it may be just the right indicator of success for a distributed computing application in which several software agents are working towards their goals, but the user of the system is only interested in (at least) one of them achieving its goal as quickly as possible. Then, if agents measure their individual welfare in terms of

how close they are to achieving their own goal, the elitist social welfare correctly reflects the value of a given state for the user of the system.

Using the terminology of Wolpert and Tumer [8], while understanding the social consequences of agents having certain behaviour profiles constitutes a *forward problem*, the design of such behaviour profiles with the aim of achieving a particular effect at the social level presents us with an *inverse problem*. Mechanism design (referred to as "inverse game theory" by Papadimitriou [9]) is an example for such an inverse problem.

Mechanism design [10, 11] is concerned with the problem of designing suitable rules of interaction between agents such that the outcome of that interaction can be guaranteed to be "optimal", given a suitable criterion for optimality and assuming certain interests on the part of individual agents. A standard example is the design of auction protocols that maximise revenue for the auctioneer and reduce the need for counter-speculation on behalf of the bidders [12]. Often, the notion of optimality can be defined in terms of a social welfare ordering. The interests of individual agents are typically taken to be fixed: agents are assumed to be *rational* in the sense of aiming solely at maximising their personal welfare. In welfare engineering, rather than designing an interaction mechanism for a given notion of social welfare and a given type of agent, we introduce a further variable into the equation by making the *rationality criteria* on the basis of which agents decide on their moves (such as whether or not to accept a proposal) a further object of design. These rationality criteria determine the *behaviour profile* of an agent. For instance, a negotiation system populated by agents that have been designed to accept deals (concerning the exchange of resources) that either benefit themselves or that are inequality-reducing can be shown (under a number of conditions) to converge towards a state that is *Lorenz optimal*, a notion of social optimality combining ideas from both the utilitarian and the egalitarian programme [1, 4].

In summary, a distributed social optimisation problem, such as the problem of finding a socially optimal allocation of resources by means of negotiation, has the following three important parameters:

(i) the *social welfare ordering* used to assess the quality of a solution;
(ii) the *social interaction mechanism* used by the agents to arrive at a solution;
(iii) the *behaviour profiles* of individual agents further restricting their moves within the boundaries given by the interaction mechanism.

Welfare economics provides several off-the-shelf solutions for (i), while welfarism is particularly concerned with assessing their respective benefits from a general point of view. Mechanism design is concerned with (ii), for a given choice regarding (i) and taking (iii) as fixed. Welfare engineering addresses all three parameters, but particularly (i) and (iii): by formulating tailor-made social welfare orderings for specific applications and by designing appropriate agent behaviour profiles (possibly in tandem with a social interaction mechanism).[3]

---

[3] The idea of working with tailor-made concepts rather than aiming for general solutions is also present in the *automated mechanism design* approach of Conitzer and Sandholm [13].

3 Resource Allocation by Negotiation

Let us consider a society of (at least 2) agents $\mathcal{A}$, and a finite set of discrete (non-divisible) resources $\mathcal{R}$. A resource allocation is a partitioning of the set $\mathcal{R}$ amongst the agents in $\mathcal{A}$. For instance, given an allocation A with $A(i) = \{r_3, r_7\}$, agent i would own resources r_3 and r_7. Given a particular allocation of resources, agents may agree on a (multilateral) *deal* to exchange some of the resources they currently hold. In general, a single deal may involve any number of resources and any number of agents. It transforms an allocation of resources A into a new allocation A'; that is, we can define a deal as a pair $\delta = (A, A')$ of allocations (with $A \neq A'$). To measure their individual welfare, every agent $i \in \mathcal{A}$ is equipped with a *utility function* u_i mapping sets of resources (subsets of $\mathcal{R}$) to rational numbers. We abbreviate $u_i(A) = u_i(A(i))$ for the utility value assigned by agent i to the set of resources it holds for allocation A.

This describes our formal negotiation framework in its most abstract form. As we shall discuss next, this framework allows for many classical concepts to be easily represented.

Monetary payments. A deal may be coupled with a number of monetary side payments to compensate some of the agents involved for an otherwise disadvantageous deal. Rather than specifying for each pair of agents how much the former is supposed to pay to the latter, we simply say how much money each and every agent either pays out or receives. This can be modelled using a *payment function* p mapping agents in $\mathcal{A}$ to rational numbers. Such a function has to satisfy the side constraint $\sum_{i \in \mathcal{A}} p(i) = 0$, *i.e.* the overall amount of money in the system remains constant. If $p(i) > 0$, then agent i *pays* the amount of $p(i)$, while $p(i) < 0$ means that it *receives* the amount of $-p(i)$. We distinguish *deals with money* and *deals without money*. For the latter, $p(i)$ is required to be 0 for every agent $i \in \mathcal{A}$. Note that for the framework without money, it would be sufficient to model an agent's preferences by means of a (not necessarily strict) total order over alternative bundles of resources.

Limited money. The model as it is allows the use of arbitrarily high side payments: each agent can give an unlimited amount of money during a deal. This is not realistic, as agents are assumed to be "infinitely rich" [2]. Another, much more realistic, way of handling money is to turn it into a resource. As our model only allows us to use discrete resources, we need to "discretise" money, by choosing the smallest money unit the system will handle (for example a euro), and by creating the corresponding resources. Thus, if we decide to put 1000 euros in the society of agents, we can choose to put 100 resources of 1 euro, and 90 resources of 10 euros each, as if it were bank notes or coins. Thus, in addition to the "normal" resources in $\mathcal{R}$ we create the set of resources

$$R_{mon} = \{r_1^{\text{€}1}, \ldots, r_{100}^{\text{€}1}, r_1^{\text{€}10}, \ldots, r_{90}^{\text{€}10}\}$$

We also have to make sure that these resources have the appropriate value. For each agent i having the resource set R, its individual welfare must be such that:

$$u_i(R) = u_i(R \backslash R_{mon}) \; + \; |R \cap \{r_1^{\text{€}1}, \ldots, r_{100}^{\text{€}1}\}| + \; 10 \times |R \cap \{r_1^{\text{€}10}, \ldots, r_{90}^{\text{€}10}\}|$$

Approximating flows. Using the same idea, it is possible to approximate the representation of continuous resources such as water or energy. For example, if a group of farmers wishes to exploit a river to irrigate their land, the water flow could be divided into 100 resources, each representing one percent of the total.

Roles. In many applications, such as most types of auctions, agents have fixed roles: some agents own resources to begin with and are sellers, while other agents have money and are buyers. This can also be represented in this framework by putting suitable restrictions on the admissibility of deals. For instance, the legality of a deal δ for a buyer i would be modelled as follows:

$$\delta = (A, A') \text{ allowed iff } A(i) \subseteq A'(i)$$

Alternatively, rather than restricting the range of legal deals, we can also model the utility functions of agents in such a way that they will behave as either sellers or buyers. For instance, if one agent values a given resource less than another agent, then the former will have an incentive to sell that resource to the latter.

Protocol restrictions. We can also express restrictions on the negotiation protocol and the agent communication language used to agree on deals. For instance, if only bilateral negotiation is possible, *i.e.* if any one deal may involve no more than two agents, this can also be modelled by means of suitable restrictions on the admissibility of a deal δ.

4   The Problem of the Designer Scope

As noted by Wurman et al. [14], *"when analyzing any multiagent involving negotiation, we must be very careful to clearly state which elements of the system are under the control of the designer"*. These authors distinguish three cases:

- *agent scope* —the designer controls a single agent.
- *mechanism scope* —the designer controls the mechanism, but not the agents that participate in it.
- *system scope* —the designer controls both the mechanism and the agents.

In addition to these cases, we can also envisage mixed situations where the designer may control the mechanism and a subset of the agents of the system. To make things clear, we give some definitions concerning the different roles of the actors in a resource allocation process:

- *proprietor role* —the person who actually owns the application and defines the mechanism. We assume that this role is taken by exactly *one* person.
- *end-user role* —the role taken by people or organisations who will use the application. Each user may own a number of agents.

Note that the roles can be cumulated, that is, the same person can have both the role of the proprietor and that of a user. This could be the case when the proprietor initially owns the resources to be distributed. It is also important to stress that there is, in theory, no requirement for an end-user to own a single agent. In most applications however, it is explicitly forbidden to hold more than one agent. This is a very important problem, considering that it is technically very difficult (or impossible) to design systems that prevent users from adopting this strategy. In e-auctions, this problem is known as the *false-name bids* problem [15], and a current trend of research is developing mechanisms that are strategy-proof regarding this issue.

When the proprietor is represented by one or several agents in the system, it can also control and modify their individual utility functions or other aspects of their behaviour profiles.

5   Example Applications

In order to further demonstrate the wide relevance of the abstract resource allocation framework presented in Section 3, we introduce three example applications, which we shall also refer to throughout the next section when we we are going to use these applications to illustrate our design criteria. Amongst these applications, only the last gives the designer a system scope; the others are cases where the designer only has mechanism scope.

E-Auctions. Different kinds of *e*-auctions have been implemented on the Internet, in the context of B2C (business-to-customer), C2C (customer-to-customer), or B2B (business-to-business) applications. Despite a first-sight similarity, C2C *e*-auctions platforms all have different characteristics which make it difficult to offer a single description. We base our discussion on three important C2C *e*-auctions platforms, namely *EmClub*, *EBay*, and *321Enchere*. In these applications, the proprietor does not necessarily hold all the resources initially (the role of the application is just to allow negotiation between users). *E*-auctions platforms have strong constraints on the type of interaction (cf. negotiation protocol). Clearly, there are sellers and buyers in the sense that the former can only sell, and the latter only buy. A famous example of B2B *e*-auctions, possibly involving combinatorial deals, is the spectrum licenses allocation process led by the Federal Communications Commission (FCC) in the United States. This allocation process involves auctioning off thousands of licenses with different geographic coverage and bandwidth. In order to deal with the large number of licences, these auctions were dispatched into several groups. The state (proprietor of the system) initially owns (all) the resources, so it may be represented by an agent in the society. The type of auction used by the FCC is the Simultaneous Multiple Round auction (SMR). This application of course necessitates that we model the roles of seller and buyer, as discussed in the previous section.

Allocation of satellite resources. Lemaître et al. [16] describe an earth observation application where users send observation requests to a satellite they have

jointly funded. Resources are earth observation images, which are initially held by the virtual proprietor (that is, the coalition of all the users: roles are cumulated). While there is the option to include a proprietor agent in the system (the "satellite agent"), closer inspection of the problem reveals that it may be unnecessary, as we are not concerned with the individual welfare of that agent.

Multiagent patrolling. To patrol is the act of walking or travelling around an area, at regular intervals, in order to protect or supervise it. This task is by nature a multi-agent task and there are a wide variety of problems that may be formulated as patrolling task. As a concrete example, during the development of an interactive computer game, one may face the problem of coordinating a group of units to patrol a given rough terrain in order to detect the presence of "enemies". The quality of the strategies used for patrolling may be evaluated using different measures. Informally, a good strategy is one that minimises the time lag between two passages to the same place and for all places. In [17], it was shown that in many applications of the patrolling problem, the territory could be represented by a graph. Given such a graph, the patrolling task refers to continuously visiting all the graph nodes so as to minimise the time lag between two visits. The edges may have different associated lengths (weights) corresponding to the real distance between the nodes. Recently [18], the patrolling problem has been formalised as a resource allocation problem. More precisely, each node of the graph to be explored can be represented by a resource, and the utility of each agent represented how well it patrols over the nodes (resources) it owns. In addition, agents can exchange nodes (resources) using a negotiation procedure, in order to maximise their patrolling performance.

6 Criteria for Social Welfare Selection

How do we assess the overall well-being of the society? There exists a large variety of social welfare measures that one can think of. To start with, there are a number of measures that have long been studied in welfare economics. On top of that, one may design new social welfare orderings that may not be appropriate in human societies, but which could be of relevance in artificial ones.

- *Utilitarian* [4, 2] —the utilitarian social welfare of an allocation of resources A is defined as the sum of utilities enjoyed by its members.
- *Nash product* [4] —the Nash product of an allocation of resources A is defined as the product of utilities enjoyed by its members.
- *Egalitarian* [4, 3] —the egalitarian social welfare of an allocation of resources A is defined as the utility enjoyed by the currently weakest agent.
- *Elitist* [1] —the elitist social welfare of an allocation of resources A is defined as the utility enjoyed by the currently happiest agent.
- *Lorenz optimality* [4, 1] —this is a combination of ideas from the utilitarian and the egalitarian approaches.
- *Envy-freeness* [19, 1] —an allocation of resource is envy-free iff there is no agent that would prefer another agent's set of resources over its own.

– *Pareto optimality* [4, 2] —an allocation of resources A is called Pareto optimal iff there is no other allocation that would make at least one of the agents in the society better off without making any of the others worse off.

A key difference between these different social welfare measures lies in the fact that some of them require *interpersonal* comparison of satisfaction levels, while others do not. Utilitarianism and egalitarianism, for instance, only make sense if we have the ability to compare utilities ascribed by different agents to their allocations. Envy-freeness and Pareto optimality, on the other hand, only require that each individual is able to compare its own alternatives.

Our aim is to present a list of criteria that should guide the designer of an application who wants to use the multiagent resource allocation framework, and in particular to support him or her in choosing the appropriate social welfare measure. The list that we are going to put forward is admittedly incomplete, but consists of the criteria we found had the most obvious consequences on the choice of a relevant social welfare measure.

6.1 Type of proprietor payment

We start by turning our attention towards an issue of critical importance during the design of a resource allocation system, namely the means by which the proprietor of the application actually gets paid. We envisage different possibilities (not mutually exclusive, as our application examples shall show):

– *Utility-dependent* —this corresponds to cases where users will contribute to the proprietor gain at a level which depends on their own satisfaction, as expressed by their utility functions. Typically that could be done by imposing of a tax on their gains.
– *Transaction-dependent* —the proprietor is payed on the basis of the sequence of transactions. For example, we may have a tax on each transaction (whatever its content), or we may have taxes on the number of resources actually exchanged, and so on. A variant of this case is time-dependence, where only the duration of the negotiation matters, regardless on the actual length (in terms of the number of transactions) of the process.
– *Membership-dependent* —the agents pay a fee when they enter the society in order to negotiate.

Apart from the purely utility-dependent case, it would then be necessary to introduce new global parameters to assess the quality of the negotiation process. In many cases, however, utility functions can be fixed such that they influence the global social welfare. Note that we do not make any assumptions as to how the reward is actually transfered to the proprietor. This can done directly from the users to the proprietor, by means of money transfer for instance. Alternatively, some authority external to the society can interfere and give rewards and penalties. This can be done on the basis of a separate agenda (for instance in the case of public services, you should make sure that everyone has some minimal access to the resources). This authority, however, would in the end base its judgement on one the items listed above.

Example applications. Each of the C2C *e*-auction applications has specific strategies regarding payment, but they generally use a mixture of transaction-dependent and membership-dependent strategies. *EBay* and *321Enchere* require that sellers pay a fee to enter the auction, and also applies a tax on each deal, which depends on the amount and on the type of object sold. Together with tax, sellers have to pay when they conclude a transaction. In fact, a fourth strategy is also used by these platforms: sellers may have the possibility to pay in order to have options facilitating the deals (advertising, photo, etc.). This is different in the case of the allocation of earth observation images. The fact that users have co-funded the satellite, may be interpreted a partners initially paying some sort of membership fee. Of course, the satellite should also be exploited in the most efficient way, so each user's satisfaction will depend upon its utilities. In the case of the spectrum allocation process led by the FCC, it is clearly a utility-dependent strategy: the (only) seller will collect at the end of the auction process the payments the buyers are committed to pay (which depend on their gains).

Discussion. When the proprietor payment is utility-based, there is a strong incentive to adopt a utilitarian framework. Transaction-dependent payments correspond to very practical cases, for instance situations where we should take into account the cost or the gain induced by a transaction. Membership-dependent payment of the proprietor require a rather different approach. As the gain enjoyed by the proprietor is actually defined at the beginning of an application run, the focus will shift to a different matter, namely making the application attractive for a large number of agents (note that this only makes sense if the application allows for multiple runs, of course). An important ingredient is then to make sure that each agent receives a fair share of the overall gain, *i.e.* in such applications we would typically favour an egalitarian notion of social welfare or we would try to achieve envy-free allocations of resources.

6.2   Application dynamics during a run

The next category of criteria that we shall investigate details how the society may evolve over time. It is first important to determine whether (and how) the number of users may vary *during* an application run. We envisage different cases:

 - *Fixed* —application users remain unchanged during an application run.
 - *Restricted* —the number of users may vary, but only under predefined conditions. These conditions may be of various kinds. First of all, there may be unidirectional restrictions: it is possible that only *new* users are allowed, or symmetrically that users are only allowed to quit the application. On an orthogonal perspective, it is possible that the restriction applies to the whole set of agents (*e.g.* the application must always involve between 10 and 20 simultaneous users), or that agents are permitted to enter the society if they fulfil certain criteria (*e.g.* holding some resources).
 - *Unrestricted* —users are allowed to enter and quit the system as they wish.

Similarly, we may also distinguish to what extent the set of resources present in the system may change during an application run.

Example applications. All C2C *e*-auctions platforms require buyers and sellers to be registered when participating in the auction (they have to create an account). New buyers can enter the auction even if it has already started, bringing along new resources as well. The FCC *e*-auction was open to any eligible company or individual that submitted an application and payment up-front, and that was deemed a qualified bidder by the Commission. To comply with the procedure, each buyer must be uniquely registered, *i.e.* there are no incoming agents.

Discussion. The application dynamics will have a direct impact on the kind of agent society used, and can be related to the well-known classification of agent societies proposed by Davidsson [20]. At first glance, users and resources migration during an application run may look somewhat beyond what our abstract framework can handle. However, although most of the theoretical results on the possibility to negotiate socially optimal allocations reported in previous papers [2, 3, 1] do not directly apply to societies where the number of agents may vary, we do not regard this as a strong limitation. Firstly, we must observe that most of the concepts used still make sense in this extended framework. Under certain restrictions, they may well be used and provide useful results. For instance, if we define utilitarian social welfare in terms of average utility rather than the sum of utilities, this concept can be used in a meaningful manner for societies with varying membership. Secondly, such extensions have been considered to some extent in the welfare economics literature [10] and it seems likely that some results could be transferred to our framework as well.

6.3 Application dynamics between runs

Under the same category, a key concern of the welfare engineer should be to consider whether the application could be run several times, and if so, whether and how the characteristics could be modified between the potential different application runs. It is indeed possible for an application to be run under a fixed policy regarding the number of users, while allowing user or resource migration between different applications runs.

A similar distinction as given above for the application dynamics during a run applies to the application dynamics *between* runs (for both users and resources).

Example applications. The satellite application can be used several times by the same users. Each negotiation phase starts with a new bundle of images to be allocated. C2C *e*-auctions applications can be run several times, but users may (and actually are likely to) be different. On the other hand, the allocation of spectrum licenses has been run only once.

Discussion. User and resource migration *between* different application runs has quite different consequences. Clearly the proprietor will be motivated to take into account long-term consequences: under the assumption that the proprietor payment is somewhat related to the number of users enjoying the application, it could for instance make sense to design the platform so that the satisfaction

of users will motivate them to join the application instead of quitting this one (and possibly joining a concurrent one), hence motivating an egalitarian flavour (even if this may decrease the proprietor's profit for a single run). Also, if the application is expected to be run several times with the same users (as a coalition), it could be important to ensure that envy will not jeopardise the coalition in the long run.

7 Concluding Remarks

In this paper, we have shown that multiagent resource allocation is a powerful paradigm which covers a wide range of applications. Following the ideas of the *welfare engineering* approach, we have also presented a number of criteria an application designer building such an application can use to decide on a suitable social welfare ordering for measuring the quality of a resource allocation.

To conclude, it is interesting to examine whether these applications actually implement the kind of measure hinted at by our discussion of the criteria. *E*-auctions have largely adopted a purely utilitarian approach, even in cases where the possible repetition of application runs with the same users should motivate the introduction of some sort of fair treatment of the users. A few years ago, studying different retailer sites, Guttman and Maes [21] already noticed that *"online auctions are unnecessarily hostile to customers and offer no long-term benefits to merchants"*, even if merchants *"care less about profit on any given interaction and care more about long-term profitability"*. In the specific case of the FCC *e*-auction, which was run only once, we also witness an utilitarian approach, as expected. C2C sites are based on different kinds of payments, and in particular on membership fees. To secure a minimum level of satisfaction of users, they have developed different strategies. One of them is the "reserve price" option: sellers can use it to stimulate bidding on their item, even if they would not sell if the price happens to be lower than their reserve price.

The satellite application requires both a fair treatment of the co-funders, and an efficient use of the satellite. This leads to the adoption of procedures involving notions of social welfare borrowing from both utilitarian and egalitarian principles. In fact, Lemaître et al. [16] propose and experiment with four different procedures to cope with this efficiency/equity tradeoff.

As far as the multiagent patrol application is concerned, the appropriate choice of a social welfare ordering really depends on the target application (video games, Internet applications, etc.). On the one hand, the utilitarian social welfare measures how well the patrolling job is done *on average* and will favour strategies in which the average time lag between two visits is to be minimised. On the other hand, the egalitarian social welfare estimates how bad the worst of the agents does, and will favour patrolling strategies in which all parts of the territory are to be visited equally often.

References

1. Endriss, U., Maudet, N.: Welfare engineering in multiagent systems. In: Engineering Societies in the Agents World IV. LNAI, Springer-Verlag (2004)
2. Endriss, U., Maudet, N., Sadri, F., Toni, F.: On optimal outcomes of negotiations over resources. In: Proceedings of the 2nd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2003), ACM Press (2003)
3. Endriss, U., Maudet, N., Sadri, F., Toni, F.: Resource allocation in egalitarian agent societies. In: Secondes Journées Francophones sur les Modèles Formels d'Interaction (MFI-2003), Cépaduès-Éditions (2003)
4. Moulin, H.: Axioms of Cooperative Decision Making. Cambridge University Press (1988)
5. Sen, A.K.: Collective Choice and Social Welfare. Holden Day (1970)
6. Rawls, J.: A Theory of Justice. Oxford University Press (1971)
7. Harsanyi, J.C.: Can the maximin principle serve as a basis for morality? American Political Science Review **69** (1975) 594–609
8. Wolpert, D.H., Tumer, K.: An introduction to collective intelligence. Technical Report NASA-ARC-IC-99-63, NASA Ames Research Center (1999)
9. Papadimitriou, C.H.: Algorithms, games, and the Internet. In: Proceedings on 33rd Annual ACM Symposium on Theory of Computing (STOC-2001), ACM (2001)
10. Arrow, K.J., Sen, A.K., Suzumura, K., eds.: Handbook of Social Choice and Welfare. Volume 1. North-Holland (2002)
11. Osborne, M.J., Rubinstein, A.: A Course in Game Theory. MIT Press (1994)
12. Vickrey, W.: Counterspeculation, auctions and competitive sealed tenders. Journal of Finance **16** (1961) 8–37
13. Conitzer, V., Sandholm, T.: Complexity of mechanism design. In: Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI-2002), Morgan Kaufmann (2002)
14. Wurman, P.R., Wellman, M.P., Walsh, W.E.: Specifying rules for electronic auctions. AI Magazine **23** (2002) 15–23
15. Yokoo, M., Sakurai, Y., Matsubara, S.: The effect of false-name bids in combinatorial auctions: New fraud in Internet auctions. Games and Economic Behaviour **46** (2004) 174–188
16. Lemaître, M., Verfaillie, G., Fargier, H., Lang, J., Bataille, N., Lachiver, J.M.: Equitable allocation of earth observing satellites resources. In: Proceedings of the 5th ONERA-DLR Aerospace Symposium (ODAS'03). (2003)
17. Machado, A., Ramalho, G., Zucker, J.D., Drogoul, A.: Multi-agent patrolling: An empirical analysis of alternative architectures. In: Proceedings of the 3rd International Workshop on Multi-agent Based Simulation (MABS-2002). LNCS, Springer-Verlag (2002)
18. Almeida, A., Ramalho, G., Santana, H., Tedesco, P., Menezes, T., Corruble, V.: Recent advances on multi-agent patrolling. In: Proceedings of the Brazilian Symposium on Artificial Intelligence. (2004)
19. Brams, S.J., Taylor, A.D.: Fair Division: From Cake-cutting to Dispute Resolution. Cambridge University Press (1996)
20. Davidsson, P.: Categories of artificial societies. In: Engineering Societies in the Agents World II. LNAI, Springer-Verlag (2003)
21. Guttman, R.H., Maes, P.: Agent-mediated integrative negotiation for retail electronic commerce. In: Agent Mediated Electronic Commerce. LNCS, Springer-Verlag (1999)