Multiagent Fair Optimization with Lorenz Dominance

(Extended Abstract)

Lucie Galand
PSL, Univ. Paris-Dauphine
LAMSADE, CNRS UMR 7243
F-75775, Paris Cedex 16,
France
lucie.galand@dauphine.fr

Thibaut Lust
Sorbonne Universités, UPMC
Univ Paris 06, CNRS, LIP6,
UMR 7606
F-75005, Paris, France
thibaut.lust@lip6.fr

ABSTRACT

This paper deals with fair optimization problems where several agents are involved. In this setting, a solution is evaluated by a vector whose components are the utility of the agents for this solution, and one looks for solutions that fairly satisfy all the agents. Lorenz dominance has been proposed in economics to refine the Pareto dominance by taking into account satisfaction inequality among the agents. The computation of Lorenz efficient solutions in multiagent optimization is however challenging (it has been shown intractable and NP-hard on certain problems). Nevertheless, to our knowledge, very few works address this problem. We propose thus in this work new methods to generate Lorenz efficient solutions. More precisely, we consider the adaptation of the well-known two-phase method proposed in biobjective optimization to the bi-agent optimization case, where one wants to directly compute the Lorenz efficient solutions. We study the efficiency of our method by applying it on the bi-agent knapsack problem.

Categories and Subject Descriptors

F.2 [**Theory of computation**]: Analysis of algorithms and problem complexity; J.4 [**Social and Behavioral Sciences**]: Economics; G.1.6 [**Numerical Analysis**]: Optimization—*Integer programming*

General Terms

Algorithms, Economics

Keywords

Multiagent optimization, Fairness, Lorenz dominance

1. INTRODUCTION

In multiagent decision problems, where one wants to make a decision with respect to the preferences of several agents, the concept of *fairness* turns out to be a crucial concern. This occurs for example when several agents (e.g. countries, companies) share the exploitation of an earth observation

satellite in order to reduce their cost [7]. When the preferences are cardinal, the agents express their preferences over the alternatives through *utility functions*, each agent having his/her own utility function to maximize. Therefore a decision is evaluated by a vector of utilities where a component represents the utility of an agent for this decision. Since there is generally not a solution that is the most preferred for all the agents, one has to determine compromise solutions that fairly satisfy all the agents. *Pareto efficiency* (*P*-efficiency) enables to define a partial preorder over the solutions based on the unanimity principle: if all the agents prefer a solution x to a solution y, then solution y is considered as dominated by solution x. However some very unfair solutions can be *P*-efficient. The *Lorenz dominance* has been proposed in economics to refine the Pareto dominance by taking into account satisfaction inequality among the agents. Roughly speaking, the Lorenz dominance enables to select all the *P*-efficient solutions that realized well-balanced compromises between the utilities of the agents. Endriss *et al.* [6] have proposed conditions under which negotiation in a multiagent system will converge to an allocation that is Lorenz efficient (*L*-efficient). Nevertheless, a few works deals with the determination of the *L*-efficient solutions (see [2, 5, 8]), which is generally a difficult problem (NP-complete and intractable [5, 8]). The aim of this work is to study the problem of multiagent fair optimization, where one looks for the *L*-efficient solutions in a combinatorial optimization problem.

2. MULTIAGENT FAIR OPTIMIZATION

2.1 Lorenz dominance

We consider in this work that p utility functions have to be maximized. The Lorenz dominance can be defined through the construction of particular vectors, called *generalized Lorenz vectors*.

DEFINITION 1. *For all* $y \in \mathbb{R}^p$, *the* generalized Lorenz vector *of* y *is the vector* $L(y)$ *defined by:*

$$L(y) = (y_{(1)}, y_{(1)} + y_{(2)}, \ldots, y_{(1)} + y_{(2)} + \ldots + y_{(p)})$$

where $y_{(1)} \leq y_{(2)} \leq \ldots \leq y_{(p)}$ *represent the components of* y *sorted by non-decreasing order.*

DEFINITION 2. *The* Lorenz dominance *relation (L-dominance for short) is defined for all* $y^1, y^2 \in \mathbb{R}^p$ *by:*

$$y^1 \succ_L y^2 \iff [L(y^1) \succ_P L(y^2)],$$

where $\succ_P$ *denotes the Pareto dominance relation* $(y^1 \succ_P y^2 \iff [\forall k \in \{1, \ldots, p\}, y_k^1 \geq y_k^2 \text{ and } y^1 \neq y^2])$.

The space in which the generalized Lorenz vectors of a solution are represented is called the *Lorenz space*. Within a feasible set $\mathcal{X}$, any element x^1 is said to be *L-efficient* (resp. *P*-efficient) when there is no x^2 in $\mathcal{X}$ such that $u(x^2) \succ_L u(x^1)$ (resp. $u(x^2) \succ_P u(x^1)$).

2.2 New methods

As *L*-efficient solutions are also *P*-efficient, one could resort to an approach that would consist in generating all the *P*-efficient solutions and then keeping only the *L*-efficient solutions. One of the most famous methods in multiobjective optimization is the *two-phase method* that enables to efficiently generate the *P*-efficient solutions to a biobjective problem (see e.g. [9]). It consists in generating first the subset of *P*-efficient solutions that optimize a weighted sum (i.e. *supported P*-efficient solutions), and second the other *P*-efficient solutions. However, the number of *L*-efficient solutions can be very small compared to the number of *P*-efficient solutions. Furthermore, the generation of *P*-efficient solutions is generally hard (see e.g. [1, 4]). For these reasons, we propose new methods that directly determine the *L*-efficient solutions. More precisely we study the adaptation of the two-phase method proposed in biobjective optimization to the fair optimization framework when two agents are involved.

Straight adaptation of the two-phase method. The adaptation of the two-phase method to generate only the *L*-efficient solutions thoroughly follows the original method, but in the Lorenz space. In the first phase, all the supported *L*-efficient solutions are generated. Actually, this amounts to optimizing Ordered Weighted Averages (OWA) [10], with decreasing positive weights, in the utility space. In the second phase, all other *L*-efficient solutions are determined, by exploring the search space defined by two consecutive *L*-efficient solutions in the Lorenz space.

Supported *P*-efficient solutions based method. Even if the straight adaptation of the two-phase method is theoretically interesting, the main drawback is in the first phase: the OWA functions that have to be optimized are non-linear and therefore even generating only the supported *L*-efficient solutions will be computationally expensive. We propose thus another method where the optimization of OWA functions is avoided. One can show that one can identify in the utility space the subset of supported *P*-efficient solutions that are *L*-efficient. It enables us to generate those *L*-efficient solutions by linear optimizations in the first phase. In the second phase, all other *L*-efficient solutions are determined, by exploring the search space defined by the *L*-efficient solutions previously determined.

3. EXPERIMENTAL RESULTS

We have applied the method based on the supported *P*-efficient solutions to the bi-agent knapsack problem. We used the instances developed by Bazgan *et al.* [3] to solve the multiobjective knapsack problem. The results are given for random (no relation between the utilities) instances (Type A, 100 items) and for instances with positive correlations (Type B, 600 items) in Tables 1 and 2. The experiments have been run on a Intel(R) Core(TM) i7-3820 CPU at 3.60GHz. We compare the running times of the method based on the supported *P*-efficient solutions (called SP) with the running times of the method of Perny *et al.* [8] (called Rkg). We report the number of the instance (from 0 to 9), the number of *P*-efficient solutions (#P) and *L*-efficient solutions (#L) and the CPU times in seconds.

#	#P	#L	CPU(s) Rkg	CPU(s) SP	#	#P	#L	CPU(s) Rkg	CPU(s) SP
0	125	6	1.98	1.40	0	66	22	/	4.56
1	184	8	22.43	2.63	1	71	17	/	2.13
2	171	6	5.76	1.43	2	111	26	/	7.76
3	198	18	/	2.49	3	88	21	/	12.32
4	112	6	0.92	0.45	4	49	20	/	8.16
5	136	5	0.17	0.16	5	82	24	/	7.05
6	135	12	117.28	0.78	6	56	25	/	15.71
7	155	49	/	979.5	7	74	23	/	69.56
8	163	4	0.34	0.24	8	97	2	3.52	0.70
9	98	7	6.70	0.44	9	49	20	/	3.13

Table 1: Type A   Table 2: Type B

We see that the *L*-efficient solutions represent only a small part of the *P*-efficient solutions. A sign "/" means that the method was not able to solve the instance within 20 minutes. We can observe that the method SP is faster.

REFERENCES

[1] H. Aziz, F. Brandt, and P. Harrenstein. Pareto optimality in coalition formation. In G. Persiano, editor, *Algorithmic Game Theory*, volume 6982 of *Lecture Notes in Computer Science*, pages 93–104. Springer Berlin Heidelberg, 2011.

[2] D. Baatar and M. Wiecek. Advancing equitability in multiobjective programming. *Computational & Applied Mathematics*, 52(1-2):225–234, 2006.

[3] C. Bazgan, H. Hugot, and D. Vanderpooten. Solving efficiently the 0-1 multi-objective knapsack problem. *Computers & Operations Research*, 36(1):260–279, 2009.

[4] M. Ehrgott. *Multicriteria Optimization. Second edition.* Springer, Berlin, 2005.

[5] U. Endriss. Reduction of economic inequality in combinatorial domains. In *AAMAS*, pages 175–182, 2013.

[6] U. Endriss, N. Maudet, F. Sadri, and F. Toni. Negotiating socially optimal allocations of resources. *J. Artif. Intell. Res. (JAIR)*, 25:315–348, 2006.

[7] M. Lemaître and G. Verfaillie. Exploiting a common property resource under a fairness constraint: A case study. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 206–211. Morgan Kaufmann Publishers, 1999.

[8] P. Perny, O. Spanjaard, and L.-X. Storme. A decision-theoretic approach to robust optimization in multivalued graphs. *Annals OR*, 147(1):317–341, 2006.

[9] E. Ulungu and J. Teghem. The two-phases method: An efficient procedure to solve biobjective combinatorial optimization problems. *Foundation of Computing and Decision Science*, 20:149–156, 1995.

[10] R. Yager. On ordered weighted averaging aggregation operators in multicriteria decision making. In *IEEE Trans. Systems, Man and Cybern.*, volume 18, pages 183–190, 1998.

Acknowledgments