CSCI 670: Theoretical Thinking Assignment 2
Non-linear Utility Functions in Negotiation

James Hale

5 December 2021

1 Problem Background

In negotiation, as non-linearities complicate the algorithmic process of finding a solution, much work assumes linear utility functions [2, 6]. However, humans rarely have linear utility functions. Salary negotiation acts as a common example to illustrate this. Consider two people, one of whom has \$10, while the other has \$1000; it seems intuitive the person with only \$10 would value \$5 more than would the person with \$1000. This implies the importance of investigating non-linear utility functions in negotiation. In fact, mathematician Daniel Bernoulli introduced non-linear utility functions, specifically dealing with currency (ducats), in *Exposition of a new theory on the measurement of risk* [1].

Various tools try to find fair negotiation proposals, acting as a mediator between two or more human users. Spliddit, developed by Goldman and Procaccia, acts as an example of such a tool; specifically, it purports to find provably fair solutions in three main areas: splitting rent, dividing goods, and sharing credit [2]. I will investigate this tool, as described in their paper *Spliddit: unleashing fair division algorithms*—specifically with respect to their tool on goods division—and how it could extend to deal with non-linear utility functions. Additionally, one can find their tool, and play with it at the following link: http://www.spliddit.org/.

2 Mathematical Formulation

Let S_i be some negotiated outcome for a negotiator i, where s_{ij} is the level of an issue j for that negotiator—e.g., if we take a simple negotiation setting in which two negotiators negotiate an allocation of apples and oranges, the issues would be the apples and oranges, and s_{ij} for $j \in \{\text{apple, orange}\}$ would be how much of that issue negotiator i received. Then let each negotiator i have a utility function for each j, which we call u_{ij}—in the non-linear case, this will be a non-linear function. Lastly, each issue has some weight w_{ij}, which indicates the relative importance of that issue j to negotiator i. We can define the total utility for a negotiator i as follows:

$$U_i(S_i) = \sum_{s_{ij} \in S_i} w_{ij} u_{ij}(s_{ij})$$

Obviously, we want to maximize global utility (which we can define as $\sum_{i \in N} U_i(S_i)$, where N is the set of negotiators) subject to some fairness objectives. Goldman and Procaccia's Spliddit, for goods division, use the following objectives: envy freeness, proportionality, and maximin share guarantee [2]. Goldman and Procaccia define these "increasingly weaker" fairness objectives [2], which I list below.

1. **Envy freeness:** No negotiator prefers another's allocation of goods, or $\forall_{i,j \in N} U_i(S_i) \geq U_i(S_j)$ more formally [2].

2. **Proportionality:** Each negotiator gets at least $1/n^{th}$ of the total available utility from their perspective, or $\forall_{i \in N} U_i(S_i) \geq U_i(S)/|N|$ more formally [2].

3. **Maximin share guarantee:** The maximin share for a negotiator i is defined as

$$MMS(i) = \max_{X_1,\ldots,X_n} \min_j U_i(X_j)$$

I.e., the minimum guaranteed value of a negotiator i, if they create n bundles $X_1, ..., X_n$ (in a case with n negotiators) and select their bundle last [2]. So, the maximin guarantee can be formally defined as $\forall_{i \in N} U_i(S_i) \geq MMS(i)$ [2].

Goldman and Procaccia also note that certain cases may not allow the satisfaction of all fairness objectives, so they attempt to satisfy the most powerful fairness objective [2]. These are just the fairness objectives Goldman and Procaccia consider, and more could be added or exchanged.

Below, let us fully define the mathematical problem, for each of these three objectives.

$$1) \max_{S_1,...,S_n} \sum_{i \in N} U_i(S_i) \quad \text{s.t. } \forall_{i,j \in N} U_i(S_i) \geq U_i(S_j)$$

$$2) \max_{S_1,...,S_n} \sum_{i \in N} U_i(S_i) \quad \text{s.t. } \forall_{i \in N} U_i(S_i) \geq U_i(S)/|N|$$

$$3) \max_{\alpha, S_1,...,S_n} \sum_{i \in N} U_i(S_i) \quad \text{s.t. } \forall_{i \in N} U_i(S_i) \geq \alpha MMS(i)$$

I.e., since we cannot guarantee there exists a solution that satisfies all constraints, the authors' try to find a solution that satisfies one of the first two constraints, and if that is infeasible, they find the highest α to fractionally satisfy the third constraint (even this cannot be guaranteed for $\alpha = 1$, however it can be fractionally guaranteed) [2].

3 Algorithmic Questions

Traditional linear programming will not work if we use non-linear utility functions. So, the following question arises: *What algorithms exist to solve optimization problems—in the same formulation as linear programming—for non-linear objective functions?*

4 State of the Art

Let us consider the algorithm used by Goldman and Procaccia. They state, first, their algorithm attempts to find the highest satisfiable fairness objective; if the first two are infeasible, the algorithm finds the greatest $\alpha > 0$ s.t. $\forall_{i \in N} U_i(S_i) \geq \alpha MMS(i)$—i.e., it satisfies the third fairness objective at some constant fraction for each negotiator [2]. Of note, the authors reference work by Kurokawa *et al.* to state α, in this case, will be at least $\frac{2}{3}$ [2, 5]. Once the algorithm finds this constraint, they simply maximize global utility (social welfare) subject to that constraint [2]. I.e., it solves the corresponding mathematical problem defined in section 2 based on the constraint it finds satisfiable. For solving the problems, Goldman and Procaccia use programs for solving mixed integer linear programs [2].

However, per the question in section 3, we must consider optimization techniques that work in non-linear scenarios too. A quick search for *non-linear programming* yielded several possibilities. Yokota *et al.* proposed genetic algorithms as a method to solve non-linear mixed integer programming problems [8]. Another work by Sahinidis proposes BARON, which uses a branch-and-bound technique on non-linear programming problems [7]. In fact, recent research still uses BARON [4]. So, perhaps some type of branch-and-bound method could be useful in approximately solving this in the non-linear case.

5 Open Questions

There are questions regarding this new latitude a human user has to express their preferences, and how they perceive it. I.e., it would be worth investigating whether a human would notice the non-linearities, and if so what effect it has. It would also be worthwhile to investigate how users could define their own non-linear utility functions, and how that would differ from Spliddit [2]. Further, another manner in which researchers have defined non-linear scenarios is by giving items conditional value [3]; e.g., an item a may have value only if one has at least one of item b, which can lead to a bumpy and non-continuous utility space. This domain is also of high importance—e.g., one may not value birthday candles at all without the cake. Ito *et al.* propose a sampling strategy in this scenario [3].

References

[1] Daniel Bernoulli. Exposition of a new theory on the measurement of risk. *Econometrica*, 22(1):23–36, 1954.

[2] Jonathan Goldman and Ariel D Procaccia. Spliddit: Unleashing fair division algorithms. *ACM SIGecom Exchanges*, 13(2):41–46, 2015.

[3] Takayuki Ito, Hiromitsu Hattori, and Mark Klein. Multi-issue negotiation protocol for agents: Exploring nonlinear utility spaces. In *IJCAI*, volume 7, pages 1347–1352, 2007.

[4] Mustafa R Kılınç and Nikolaos V Sahinidis. Exploiting integrality in the global optimization of mixed-integer nonlinear programming problems with baron. *Optimization Methods and Software*, 33(3):540–562, 2018.

[5] David Kurokawa, Ariel D. Procaccia, and Junxing Wang. Fair enough: Guaranteeing approximate maximin shares. *J. ACM*, 65(2), feb 2018.

[6] Johnathan Mell and Jonathan Gratch. Iago: interactive arbitration guide online. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pages 1510–1512, 2016.

[7] Nikolaos V Sahinidis. Baron: A general purpose global optimization software package. *Journal of global optimization*, 8(2):201–205, 1996.

[8] Takao Yokota, Mitsuo Gen, and Yin-Xiu Li. Genetic algorithm for non-linear mixed integer programming problems and its applications. *Computers Industrial Engineering*, 30(4):905–917, 1996.