PhD Research Project Extended Abstract
Utilitarian Combinatorial Assignment

Fredrik Präntare[0000−0002−0367−2430] and Fredrik Heintz[0000−0002−9595−2471]

Linköping University
581 83 Linköping, Sweden
{firstname.lastname}@liu.se

Keywords: Combinatorial assignment · Multi-agent systems · Coalition structure generation · Optimization · Operations research.

A major problem in computer science is that of designing cost-effective, scalable *assignment* algorithms that seek to find a *maximum weight matching* between the elements of (e.g., two) sets. We consider a highly challenging problem of this type—namely that of black-box *utilitarian combinatorial assignment* (UCA), in which indivisible elements (e.g., goods/courses) have to be distributed in bundles (i.e., partitioned) among a set of other elements (e.g., buyers/students) to maximize a notion of utilitarian social welfare. This is a central problem in both artificial intelligence, operations research, and algorithmic game theory; with applications in optimal task/resource allocation [1], winner determination for combinatorial auctions [6], and team/coalition formation [3]. More formally, the UCA problem is defined as follows:

Input: A set of elements $A = \{a_1, ..., a_n\}$, a set of alternatives $T = \{t_1, ..., t_m\}$, and a ("queryable black-box") function $\boldsymbol{v} : 2^A \times T \mapsto \mathbb{R}$ that maps a value to every possible pairing of a bundle $C \subseteq A$ (also called a *coalition*) to an alternative $t \in T$.

Output: A set $\{C_1, ..., C_m\}$ with $C_i \cap C_j = \emptyset$ for all $i \neq j$ and $\bigcup_{i=1}^{m} C_i = A$, that maximizes $\sum_{i=1}^{m} \boldsymbol{v}(C_i, t_i)$.

However, UCA is computationally hard. The state-of-the-art can only compute solutions to problems with severely limited input sizes—and due to Sandholm [5], we expect that no polynomial-time approximation algorithm exists that can find a feasible solution with a provably good worst-case ratio (i.e., where the optimal solution is guaranteed to be within a multiplicative factor of the returned solution).

Moreover, as a lower bound, we need to enumerate all the $m2^n$ possible values of the value function to make any worst-case guarantee on the quality of a feasible solution. In other words, to find a solution guaranteed to be within any bound from optimum, we first need to scan the whole input and evaluate every possible pairing of a coalition $C \subseteq A$ to an alternative $t \in T$.

With this in mind, it would be interesting to investigate if, when and how low-complexity algorithms can generate feasible solutions of high-enough quality for problems with large-scale inputs and limited computation budgets. Deducing this is especially important for real-world problems, since they are often characterized by large input sizes and the need for quickly generating feasible solutions of high quality.

Against this background, this project's main contributions, that advances the state-of-the-art, are the following:

– **[Published]** We developed and benchmarked an anytime branch-and-bound algorithm for UCA. Our benchmarks show that it greatly outperforms CPLEX—an industry-grade solver by IBM. Moreover, this algorithm has subsequently been used in the commercial strategy game Europa Universalis 4 to deploy armies to different geospatial regions. [1,4,2,3]
– **[Submitted]** We developed and benchmarked two dynamic programming algorithms for UCA: one based entirely on dynamic programming, and one anytime hybrid approach that uses branch-and-bound together with dynamic programming. Our benchmarks show that our hybrid algorithm greatly outperforms CPLEX, pure dynamic programming, and all other approaches in all of our benchmarks. For example, when solving one of our most difficult synthetic problem sets, our hybrid approach found optimum in roughly 0.1% of the time that the previously best method needed, and it generated 98% efficient interim solutions in milliseconds in all of our anytime benchmarks; a considerable improvement over all previous methods.
– **[Work in progress]** We developed and benchmarked several heuristic search and Monte Carlo methods for large-scale UCA. These algorithms quickly find feasible solutions, and they find near-optimal solutions in all of our synthetic benchmarks.
– **[Work in progress]** We developed a heuristic search method for UCA that uses general-purpose heuristics together with deep neural networks to guide search.

For future work, it would be interesting to further investigate when and how machine learning can be applied to find better feasible solutions, and also benchmark our algorithms and methods on real-world problem sets.

References

1. Präntare, F.: Simultaneous coalition formation and task assignment in a real-time strategy game. In: Master thesis (2017)
2. Präntare, F., Heintz, F.: An anytime algorithm for simultaneous coalition structure generation and assignment. In: International Conference on Principles and Practice of Multi-Agent Systems. pp. 158–174 (2018)
3. Präntare, F., Heintz, F.: An anytime algorithm for optimal simultaneous coalition structure generation and assignment. Autonomous Agents and Multi-Agent Systems **34**(1), 1–31 (2020)
4. Präntare, F., Ragnemalm, I., Heintz, F.: An algorithm for simultaneous coalition structure generation and task assignment. In: International Conference on Principles and Practice of Multi-Agent Systems. pp. 514–522. Springer (2017)
5. Sandholm, T.: Algorithm for optimal winner determination in combinatorial auctions. Artificial intelligence **135**(1-2), 1–54 (2002)
6. Sandholm, T., Suri, S., Gilpin, A., Levine, D.: Winner determination in combinatorial auction generalizations. In: Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1. pp. 69–76 (2002)