Modeling Collective Constraint Optimization

Alexander Schiendorfer[1,3], Guido Tack[2], and Alexander Knapp[1]

[1] Institute for Software & Systems Engineering, University of Augsburg, Augsburg, Germany {schiendorfer,knapp}@isse.de
[2] Department of Data Science and Artificial Intelligence, Monash University, Melbourne, Australia guido.tack@monash.edu
[3] Technische Hochschule Ingolstadt, Ingolstadt, Germany

Abstract. Many real-world constraint problems involve different types of objectives belonging to multiple stakeholders. Aggregating these objectives in a socially desirable way while balancing this with other encompassing goals is challenging. Instead of doing this ad-hoc for new problems, we propose to develop a modeling framework on top of MiniZinc which consists of a reusable library of constraints and search procedures inspired by social choice theory. We highlight these aspects using an illustrative case study and give an outlook on a future research and development roadmap.

Keywords: Constraint Optimization · Social Choice · Fairness.

1 Constraint Optimization with Multiple Stakeholders

Increasingly, systems need to make decisions that involve multiple stakeholders (people, organizations, etc.) competing over shared resources. At the core of these systems lie constrained optimization problems that are subject to a number of preferences, such as shared mobility concepts [14] where collective rides have to be allocated to passengers, manufacturing as a service [16] where shared machines need to be arbitraged among manufacturers, or smart grids [15] that schedule coordinated demand times to avoid peak loads. All of these applications potentially bear many hard combinatorial problems regarding feasibility, on top of which we have to make socially acceptable decisions regarding a number of (potentially conflicting) objectives. But “socially acceptable” may involve aspects such as “fairness”, “balanced allocations”, or “least-happy maximization” and clearly depends on the application context. Ideally, we would be able to *model* what it means to be socially acceptable in a way that is transparent and comprehensible to developers and end-users alike.

In classical constraint optimization, the task is usually to pick a solution $\boldsymbol{s}$ from a constrained set of feasible solution candidates $\mathcal{S}$ that is minimal or maximal with respect to an objective function $f : \mathcal{S} \to (F, \leq_F)$ where F is an ordered set (most often the natural/real numbers with their usual ordering). Constraint programming (CP), as a particular constraint solving technology, has focused strongly on expressive constraints that – in addition to offering efficient domain filtering by virtue of propagators – lead to concise models that can serve as high-level problem specifications for end-users. We, therefore, believe that the field is well-equipped to handle challenges that arise in *collective* constraint problems. In this paper, we aim at demonstrating the need for more sophisticated collective models. We do so by means of providing an illustrative example and a preliminary approach based on incorporating social choice functions into optimization problems using the MiniZinc technology.

Related Work

The potentially symbiotic relationship between collective decision making and constraint reasoning is not new, for example, it was pointed out by Rossi in 2014 [12], most notably leading to a sequential voting procedure [2, 1] to connect social choice theory and constraint solving. What has changed is that the CP community has reached a higher level of maturity in its constraint solvers and, especially, its modeling languages such as OPL [6], Essence [5], or MiniZinc [9]. We believe that the field is ready to take on more involved problems that go beyond proof-of-concept instances or novel ad-hoc voting-based solvers. But to achieve this, it is imperative to *leverage existing solvers* and formulate socially desirable concepts on top of them, as done prototypically in [13].

Of course, pure multi-criteria optimization deals with multiple objectives, most notably via finding Pareto frontiers or minimizing distances to Utopian points [3]. Every stakeholder's preference needs to be encoded as one of the utility functions. Solving for a *large number* of objectives, however, leads to very indecisive optimization criteria [13].

Finally, while the distributed constraint optimization (DCOP) community has produced remarkable results in transferring search and propagation algorithms to distributed computing environments [4], there has been little discussion about *what to optimize for* other than a sum of utilities.

2 A New Modeling Framework

We believe that many collective constraint problems require a mix of various types of objectives, such as:

1. "classical" numeric objectives such as cost, efficiency, time, makespan;
2. Balance/fairness constraints that assert comparable criteria such as night shifts per month, distance to the closest school, or fair allocation of resource time in shared manufacturing. Fairness can amount to *equal* treatment or other criteria such as progressive/regressive taxation;
3. subjective opinions, preferences that are hard to quantify or hard to objectively compare: "I like a proposed car ride more than another one if it is a certain brand", or, "I'd rather have a direct connection to the central station than a combination of multiple short trips."

These different types of objectives require different approaches for modeling and solving. Numeric objectives can be treated directly by optimization solvers. Fairness and equity approaches have received quite a bit of attention in the literature [8], and some of them can be mapped onto standard optimization procedures. Aggregating incomparable opinions is probably best suited for voting mechanisms borrowed from social choice.

We envision a modeling and solving framework that can *combine* these objectives, depending on the specific problem at hand. Let us concretize these ideas using an example.

Designing Joint Travel Itineraries

Consider the following case study. A travel company needs to design tour plans for its customers. Each customer wants to visit certain sites of interest over a period of time (e.g., a week), and the travel company needs to pool customers into groups and organize bus tours to the sites. The stakeholders in this example are the company, touristic site owners, bus operators, and travelers.

The travel company operates under the following constraints:

– Site availability (opening ours, time windows, group size restrictions)
– Fleet size (number of buses available)
– Travel times between hotels and sites of interest
– Contractual obligations with certain sites (e.g., a minimum number of visits per year)

We can instantiate the above classes of objectives:

– There are obvious monetary costs involved. A travel company might simply be interested in maximizing their profits. But optimizing tours only for profit and ignoring the customers' preferences leads to a negative feedback loop via bad reviews and customer dissatisfaction. A driving cost factor could be the number of buses needed to trade off capacity constraints and consumer satisfaction.
– With respect to balancing and fairness, we see two possible goals: i) regarding our tour guides or sites, we might have to ensure that the workload (i.e., the rosters) is spread evenly perhaps over multiple time periods; ii) for our travelers, there could be comparable metrics such as the number of feasible sites (e.g., a handicapped person should not have to only choose among incompatible options; similarly, a family could not be able to enter a too violent museum).
– There are hard-to-quantify subjective preferences such as "I'd rather pick site a than b but this holds only if person C joins me" or "I want to be at Golden Gate only in the morning - I don't care about the afternoon". Such conflicts need to be resolved, e.g., via voting since the outcome should be an acceptable compromise for the whole group.

It is evident that all aspects affect the decision making, but no sufficiently powerful algorithmic toolkit is available. The interplay leads to several possible algorithmic instantiations. CP, in general, can help us to identify feasible solutions in terms of balance/fairness constraints, respecting other objective functions as well. For instance, the travel agency might offer only itineraries that achieve a certain amount of quota satisfaction of site owners. Resource constraints on the available buses or tour guides can enter as objectives or hard constraints. For comparable metrics concerning the balance of tour guides and sites, Rawlsian approaches could be needed [10] (e.g., when proposing an alternative solution, assure that the worst dissatisfaction is improved; the *difference principle*). The latter immediately translates into executable constraints to be added to a model.

Voting in a social-choice sense needs to be used when we fail to propose other, more objective, decision criteria.

3 Algorithm Sketches

A simple algorithm for solving this problem could proceed as follows:

1. Define and specify suitable minimal fairness/balance constraints that are acceptable as ground rules.
2. Solve the problem (including basic fairness constraints) using only the cost objective, in order to find a lower bound for the cost.
3. Define an upper acceptable cost bound. E.g., you may be prepared to accept a solution that is up to 10% more expensive than the lowest cost one, so that the other objectives are not too tightly constrained.
4. Optimize according to a social choice function defined by the users' (incomparable) preferences over the so-constrained solution set. Pick a choice set C of multiple solutions that is likely to be equivalent in terms of these preferences (e.g., all approximate Copeland winners).
5. Optimize according to the cost objective to find the best option among all the solutions in C (which won the voting).

Most of these steps, with the exception of step 4, can be implemented quite easily on top of existing modeling frameworks. In order to support the development of models and algorithms like this one, we propose to extend MiniZinc with the following features:

- A *library* of constraints that can be used to express fairness and balance conditions.
- A *meta-search* framework (based on the concepts developed for MiniSearch [11]) to enable the implementation of iterative solving algorithms.
- A *library* of constraints and meta-search algorithm building blocks for implementing social choice functions.

As demonstrated by the previous MiniSearch approach, the needed capabilities can, in principle, be implemented on top of any constraint solver, therefore leveraging the significant advances in solving technology.

For the social choice optimization from step 4, we propose to experiment with an approximative, iterative algorithm that could be considered an extension of Large Neighbourhood Search. The algorithm starts from an initial set of solutions (such as a diverse set [7] of close to optimal solutions from step 3). All agents then vote on these solutions, using social choice functions such as Copeland or Borda scores – essentially counting the number of won pairwise majority "duels" between competing solutions. That step extends simple Condorcet mechanisms presented in [13]. From the outcome of the voting, we add constraints and modify the objective for the next iteration such that it favors solutions that are *different* from the bottom-ranked solutions, and *similar* to the top-ranked ones. In combination with diversity maximization, that procedure leads to a coarse-to-fine progression where agents first negotiate the basic direction (e.g. "Italian" or "Asian" restaurant) before working out the details (e.g. "which Italian restaurant"). Preliminary experiments show that this approach can find solutions that are close to the overall Copeland or Borda winner while only generating a fraction of the overall number of solutions.

4 Conclusion

We believe that a combination of classic optimization techniques, high-level modeling constructs for balance and fairness, and social choice objectives implemented via a meta-search framework will provide a useful test bed as well as potentially a real-world framework for solving rich, constrained, multi-objective and collective constraint optimization problems. We welcome any comments on this approach and invite collaboration from anyone interested in this area.

References

1. Cornelio, C., Pini, M.S., Rossi, F., Venable, K.B.: Multi-agent Soft Constraint Aggregation via Sequential Voting: Theoretical and Experimental Results. Autonomous Agents and Multi-Agent Systems **33**(1-2), 159–191 (2019)
2. Dalla Pozza, G., Rossi, F., Venable, K.B.: Multi-agent Soft Constraint Aggregation: A Sequential Approach. In: Proc. 3rd Intl. Conf. Agents and Artificial Intelligence (ICAART'11. vol. 11 (2010)
3. Ehrgott, M.: Multicriteria Optimization, vol. 491. Springer Science & Business Media (2005)
4. Fioretto, F., Yeoh, W., Pontelli, E., Ma, Y., Ranade, S.J.: A Distributed Constraint Optimization (DCOP) Approach to the Economic Dispatch with Demand Response. In: Proc. 16th Intl. Conf. Autonomous Agents and Multiagent Systems (AAMAS'17). pp. 999–1007. International Foundation for Autonomous Agents and Multiagent Systems (2017)
5. Frisch, A.M., Harvey, W., Jefferson, C., Martínez-Hernández, B., Miguel, I.: Essence: A Constraint Language for Specifying Combinatorial Problems. Constraints **13**(3), 268–306 (2008)
6. van Hentenryck, P.: The OPL Optimization Programming Language. MIT Press (1999)

7. Ingmar, L., de la Banda, M.G., Stuckey, P.J., Tack, G.: Modelling diversity of solutions. In: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020. pp. 1528–1535. AAAI Press (2020), https://aaai.org/ojs/index.php/AAAI/article/view/5512
8. Karsu, Ö., Morton, A.: Inequity averse optimization in operational research. Eur. J. Oper. Res. **245**(2), 343–359 (2015). https://doi.org/10.1016/j.ejor.2015.02.035, https://doi.org/10.1016/j.ejor.2015.02.035
9. Nethercote, N., Stuckey, P.J., Becket, R., Brand, S., Duck, G.J., Tack, G.: MiniZinc: Towards a standard CP modelling language. In: Proc. 13th Intl. Conf. Principles and Practice of Constraint Programming (CP'07). Lect. Notes Comp. Sci., vol. 4741, pp. 529–543. Springer (2007)
10. Rawls, J.: A theory of justice. Harvard university press (1971)
11. Rendl, A., Guns, T., Stuckey, P.J., Tack, G.: MiniSearch: A Solver-independent Meta-search Language for MiniZinc. In: Proc. 21st Intl. Conf. Constraint Programming (CP'15). Lect. Notes Comp. Sci., vol. 9255, pp. 376–392 (2015)
12. Rossi, F.: Collective decision making: a great opportunity for constraint reasoning. Constraints **19**(2), 186–194 (2014)
13. Schiendorfer, A., Reif, W.: Reducing bias in preference aggregation for multiagent soft constraint problems. In: International Conference on Principles and Practice of Constraint Programming. pp. 510–526. Springer (2019)
14. Shaheen, S., Cohen, A., Zohdy, I., et al.: Shared mobility: current practices and guiding principles. Tech. rep., United States. Federal Highway Administration (2016)
15. Siano, P.: Demand response and smart grids—a survey. Renewable and sustainable energy reviews **30**, 461–478 (2014)
16. Zhang, L., Luo, Y., Tao, F., Li, B.H., Ren, L., Zhang, X., Guo, H., Cheng, Y., Hu, A., Liu, Y.: Cloud manufacturing: a new manufacturing paradigm. Enterprise Information Systems **8**(2), 167–187 (2014)