ECAI 2020
G.D. Giacomo et al. (Eds.)

doi:10.3233/FAIA200441

Efficient Allocations in Constant Time: Towards Scalable Solutions in the Era of Large Scale Intelligent Systems

Panayiotis Danassis and **Boi Faltings** [1]

Abstract. The next technological revolution will be interwoven to the proliferation of intelligent systems. As we bridge the gap between physical and cyber worlds, we will give rise to *large-scale*, multi-agent based technologies. A key challenge that cities of the future will have to face is coordination in the use of limited resources, central to which is finding an optimal allocation between agents. To truly allow for scalable solutions, we needs to shift from traditional approaches, to multi-agent solutions, ideally run *on-device*.

We present a novel heuristic (ALMA), which exhibits such properties, for solving the assignment problem. ALMA is decentralized, requires only partial feedback, and has *constant* in the total problem size running time, under reasonable assumptions on the preference domain of the agents, making it ideal for an *on-device* implementation. We have evaluated ALMA in a variety of scenarios including synthetic and *real* data, time constraints, and on-line settings. In all of the cases, ALMA was able to reach high social welfare, while being orders of magnitude faster than the centralized, optimal algorithm.

1 Introduction

One of the fundamental problems in multi-agent systems is finding an optimal allocation between agents, i.e. solving the assignment problem [5]. Example applications include role allocation, task assignment, resource allocation, etc.

Many assignment problems occur in unboundedly large settings, like for example resource allocation in urban environments (matching electric vehicles to charging stations, taxi drivers to passengers, etc.). Algorithms that solve the assignment problem, whether centralized or distributed, require runtime that depends on the problem size. Thus, as the size increases, we need to artificially cut the problem by placing bounds or spatial constraints in order to make it tractable.

A second challenge emerges from the information-restrictive nature of real-world applications. There exists a variety of decentralized algorithms for solving the assignment problem, all of which, though, require polynomial in the problem size number of messages. However, inter-agent interactions often repeat no more than a few hundreds of times. Moreover, sharing plans and preferences creates high overhead, and there is often a lack of responsiveness and/or communication between the participants [6]. Achieving fast convergence and high efficiency in such information-restrictive settings is extremely challenging.

Humans on the other hand, are able to routinely and robustly coordinate in similar everyday scenarios, often with no explicit communication. One driving factor that facilitates human cooperation is behavioral conventions [4, 1]. Inspired by human behavior, we have developed a heuristic (ALMA [2]) which is modeled on an *altruistic* convention. A distinctive characteristic of ALMA is that agents make decisions locally, based on the contest for resources that *they* are interested in, and the agents that are interested in the *same* resources. If each agent is interested in only a *subset* of the total resources, ALMA converges in time polynomial in the maximum size of the subsets; not the total number of resources. Crucially, in the realistic case where the aforementioned quantities are bounded independently of the total number of agents/resources, the convergence time remains *constant* as the total problem size increases. The latter is true by default in many real-world applications (e.g., resource allocation in urban environments), since agents only have a local (partial) knowledge of the world, and there is typically a cost associated with acquiring a resource (e.g., a taxi driver would not be willing to drive to the other end of the city to pick up a passenger). Thus, ALMA utilizes a natural characteristic of the application domain, where other algorithms (e.g., the optimal centralized solution) would require time polynomial in the total number of agents/resources due to interdependencies. The lightweight nature of ALMA coupled with the lack of inter-agent communication, and the highly efficient allocations, make it ideal for an *on-device* solution for large-scale intelligent systems (IoT devices, intelligent infrastructure, autonomous vehicles, etc.).

2 ALMA: ALtruistic MAtching Heuristic

The assignment problem consists of finding a maximum weight matching in a weighted bipartite graph, $\mathcal{G} = \{\mathcal{N} \cup \mathcal{R}, \mathcal{E}\}$. Let $\mathcal{N} = \{1, \dots, N\}$, $\mathcal{R} = \{1, \dots, R\}$ denote the set of agents, resources respectively, and let the weight of an edge $(n, r) \in \mathcal{E}$ represent the utility ($u_n(r) \in [0, 1]$) of acquiring resource r. Each agent can acquire at most one resource, and each resource can be assigned to at most one agent. The goal is to maximize the social welfare (sum of utilities), i.e., $\max_{\mathbf{x} \geq 0} \sum_{(n,r)\in\mathcal{E}} u_n(r) x_{n,r}$, subject to $\sum_{r|(n,r)\in\mathcal{E}} x_{n,r} = 1, \forall n \in \mathcal{N}$, and $\sum_{n|(n,r)\in\mathcal{E}} x_{n,r} = 1, \forall r \in \mathcal{R}$.

2.1 Learning Rule

Each agent sorts his available resources ($\mathcal{R}^n \subseteq \mathcal{R}$) in decreasing order of utility ($r_1, r_2, \dots, r_i, r_{i+1}, \dots, r_{R^n}$). The set of available actions is denoted as $\mathcal{A} = \{Y, A_{r_1}, \dots, A_{r_{R^n}}\}$, where Y refers to yielding, and A_r refers to accessing resource r. Each agent has a strategy (g_n) that points to a resource and it is initialized to the most preferred one. Alg. 1 presents the pseudo-code of ALMA.

[1] École Polytechnique Fédérale de Lausanne (EPFL),Artificial Intelligence Laboratory,Switzerland, email: {panayiotis.danassis, boi.faltings}@epfl.ch

Algorithm 1 ALMA: Altruistic Matching Heuristic [2].

```
Require: Sort resources ($\mathcal{R}^n \subseteq \mathcal{R}$) in decreasing order of utility
    $r_1, r_2, \dots, r_i, r_{i+1}, \dots, r_{R^n}$.
Require: Initialize $g_n \leftarrow A_{r_1}$, and $r_{\text{prev}} \leftarrow r_1$.
 1: procedure ALMA
 2:     while True do
 3:         if $g_n = A_r$ then
 4:             Agent $n$ attempts to acquire $r$. Set $r_{\text{prev}} \leftarrow r$.
 5:             if Collision($r$) then
 6:                 back-off (set $g_n \leftarrow Y$) with prob. $P_n(r)$.
 7:             else break.
 8:         else ($g_n = Y$)
 9:             $n$ monitors $r \leftarrow S_n(r_{\text{prev}})$. Set $r_{\text{prev}} \leftarrow r$.
10:             if Free($r$) then set $g_n \leftarrow A_r$.
```

In short, ALMA works as follows: Every agent runs Alg. 1 independently and in parallel. While contesting for a resource, each agent will back-off with probability that depends on their *own utility loss* of switching to their respective remaining resources (e.g., let $loss_n^i = u_n(r_i) - u_n(r_{i+1})$, then agent n backs-off with probability $P_n(r) = 1 - loss_n^i$). Agents that do not have good alternatives will be less likely to back-off and vice versa. If an agent backs-off, it selects an alternative resource and examines its availability (e.g., go to the next best resource, i.e. $S_n(r_{\text{prev}}) = r_{\text{prev}+1}$). We assume that agents can observe feedback from their environment (either with sensors, or a single 1-bit feedback from the selected resource). This is used to inform collisions and detect free resources. No communication between the participating agents is needed.

2.2 Convergence Time

We assume that each agent n is interested in a subset of the total resources, i.e., $\mathcal{R}^n \subset \mathcal{R}$, thus at each resource there is a bounded number of competing agents N^r. In [2] we have proven that the expected number of steps any individual agent requires to converge is independent of the total problem size (i.e., N and R); in other words, the convergence time is *constant* in these quantities [2].

2.3 Evaluation

We evaluated ALMA in a plethora of scenarios including synthetic and *real data*, time constraints, and on-line settings. In all of the cases, ALMA was able to reach high social welfare, while being orders of magnitude faster than the centralized, optimal algorithm [2].

Let us consider a Cartesian map representing a city on which are randomly distributed vehicles and charging stations. The utility received by a vehicle n for using a charging station r is proportional to the inverse of their distance, i.e., $u_n(r) = 1/d_{n,r}$. If we bound the maximum number of resources each agent is interested in, and the the maximum number of agents competing for a resource, specifically $R^n = N^r \in \{8, 16, 32, 64, 128\}$, Fig. 1 (top) shows that the average time-steps until an agent successfully claims a resource remains constant as we increase the total problem size (R, N). Compared to the Hungarian algorithm, ALMA requires approximately 7 orders of magnitude less computation time, and this number would grow boundlessly as we increase N, R. Moreover, Fig. 1 (bottom) shows that ALMA is able to reach high quality matchings, with only

[2] The convergence time is polynomial in the maximum number of desired resources, $\max_{n' \in \cup_{r \in \mathcal{R}^n} \mathcal{N}^r} R^{n'}$, and competing agents per resource, $\max_{r \in \mathcal{R}^n} N^r$.

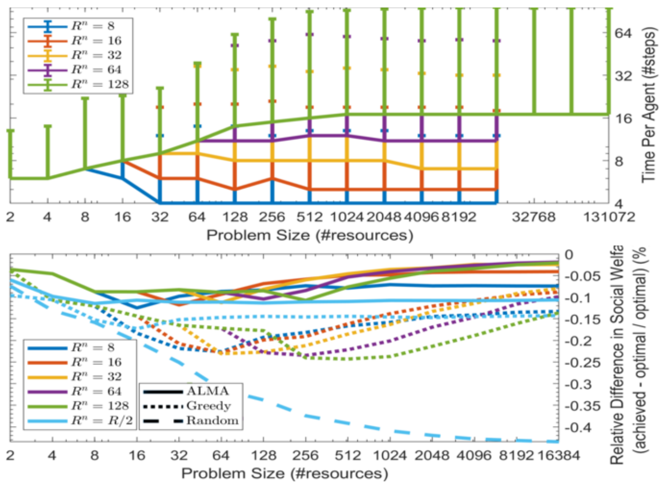

Figure 1: (top) Average time (#steps) for an agent to acquire a resource (double log), (bottom) Relative difference in SW compared to the optimal (%) (single log), for increasing R, and $N = R$ [2].

$1.9 - 12\%$ loss in SW compared to the optimal, while the greedy algorithm losses $8.0 - 24\%$ and the random algorithm $7.3 - 43.4\%$.

We have also evaluated ALMA in an *on-line* setting, involving dynamic ridesharing, and fleet relocation [3]. We used *real* data of taxi rides in New York City, and compared against 12 different algorithms over 12 metrics. In spite of the unpredictability of the on-line setting, and the dynamic nature of the demand, ALMA is consistently able to exhibit high performance – on par with the best performing, computationally heavy, centralized algorithms. Moreover, it facilitates simple relocation schemes which improve several Quality-of-Service metrics by more than 50% in certain cases; thus offering an efficient, *on-device*, end-to-end solution.

3 Conclusion

As the number and diversity of autonomous agents grows, so does the need for developing multi-agent based technologies that can run *on-device*, and produce high quality solutions. In this paper, we present such a simple decentralized heuristic for weighted matching (ALMA). ALMA uses only local information and partial feedback from the environment, to reach high efficiency allocations in *constant* in the total problem size running time. The faster convergence can outweigh the loss in utility, and allow agents robustly coordinate in large scale and under dynamic and unpredictable demand.

REFERENCES

[1] Panayiotis Danassis and Boi Faltings, 'Courtesy as a means to coordinate', in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '19, (2019).

[2] Panayiotis Danassis, Aris Filos-Ratsikas, and Boi Faltings, 'Anytime heuristic for weighted matching through altruism-inspired behavior', in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 215–222, (2019).

[3] Panayiotis Danassis, Marija Sakota, Aris Filos-Ratsikas, and Boi Faltings, 'Putting ridesharing to the test: Efficient and scalable solutions and the power of dynamic vehicle relocation', *ArXiv: 1912.08066*, (2019).

[4] David Lewis, *Convention: A philosophical study*, 2008.

[5] James Munkres, 'Algorithms for the assignment and transportation problems', *J. of the society for industrial and applied mathematics*, (1957).

[6] Peter Stone, Gal A. Kaminka, Sarit Kraus, and Jeffrey S. Rosenschein, 'Ad hoc autonomous agent teams: Collaboration without precoordination', in *Twenty-Fourth AAAI Conf.*, (2010).