Synergies Between Evolutionary Computation and Multiagent Reinforcement Learning: the Benefits of Exchanging Solutions

Ana L. C. Bazzan
Instituto de Informática, UFRGS
Caixa Postal 15064, 91.501-970 Porto Alegre, RS, Brazil
bazzan@inf.ufrgs.br

ABSTRACT

In many real-world situations in which resources are scarce, aligning the optimum of the system with the optimum of agents can be conflicting. For instance, in traffic assignment, the system's and the agents' welfare may not be aligned. In order to deal with this, in this paper a new approach is proposed, based on a synergy between: (i) a global optimization process in which the traffic authority employs metaheuristics, and (ii) reinforcement learning processes that run at each individual driver agent. Both the agents and the system authority exchange solutions that are incorporated by the other party in order to come up with an assignment of routes.

CCS CONCEPTS

• **Computing methodologies → Multi-agent systems**;

KEYWORDS

Reinforcement Learning, Evolutionary Computation, Multiagent Systems

ACM Reference format:
Ana L. C. Bazzan. 2017. Synergies Between Evolutionary Computation and Multiagent Reinforcement Learning: the Benefits of Exchanging Solutions. In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 2 pages.
DOI: http://dx.doi.org/10.1145/3067695.3075970

1 INTRODUCTION

In many real-world problems there is a conflict between the desired performance of the system as a whole, and the performance that its individual components can achieve. For an example, take congestion games: while a central authority is interested in optimizing the *average* travel time, drivers are interested in optimizing their own individual travel times. Therefore, a synergy between these two views of an optimization problem may make sense.

In the context of optimization and multiagent learning, the literature reports some works that deal with such synergy. Bazzan and Chira [1] have proposed a hybrid approach between a genetic algorithm (GA) and Q-learning (QL) and applied to the traffic assignment problem (TAP). However, the situation in which *both* the central authority and the agents can benefit was not explored. In contrast to the work in [1], in the present paper not only the central authority benefits but also the individual agents. This novel approach addresses systems in which there are individual agents competing for resources. Similarly, in [2], co-evolution is used for cooperative agents to achieve some system objective.

In short, the novel approach proposed here is based on a synergy between metaheuristics and multiagent RL. Moreover, the latter is able to deal with thousands of agents learning to use scarce resources. This task is far from solved in multiagent systems because convergence guarantees do not hold when more than one agent is learning simultaneously. Assuming that there is some sort of central authority that aims at regulating the system or at incentivizing individuals to take certain actions, our approach shows that an exchange of information can improve the performance of the overall system, as well as the performance of individual agents.

2 METHODS AND RESULTS

In a nutshell, the proposed approach is based on an algorithm (called GA<->QL) that works by biasing solutions that are computed both at agent level as well as at central authority level. In the former case, the learning task is biased by a solution coming from the central authority. In the opposite direction, the solutions to be evolved by the central authority are biased by a solution that is assembled using the agents' learned actions.

The input to the algorithm is: a set $\mathcal{A}$ of agents (each with a set of actions K^i); a (domain dependent) description of the environment (f_i, a function that gives agent A^i its reward depending on the actions of other agents, f_c, a function that gives the objective to be optimized at global level); a RL method such as QL (with its parameters' values); a population based metaheuristic such as GA (with its parameters such as mutation rate m and crossover c); and Δ, the frequency with which solutions are exchanged. The output is an element of $\times K^i \in \mathcal{K}$, i.e., a set of actions, where $k_i \in K^i$ is an action for A^i.

Given the input, an initial population of solutions is generated (for the metaheuristic), where an individual in this population is a list of size n containing an action for each agent A^i. In each learning episode (for the RL), either agents learn by interacting with the environment, or agents select an action that is recommended by the central authority. In both cases, the agents observe their rewards, update the value of their actions, and each informs its action to the central authority. This assembles a candidate solution that replaces its worst solution in the population. Then reproduction, crossover and mutation happens and the best solution is selected, which will eventually be recommended to the agents. This loop is repeated until some criteria is reached.

GECCO '17 Companion, Berlin, Germany
 978-1-4503-4939-0/17/07…$15.00
DOI: http://dx.doi.org/10.1145/3067695.3075970

Table 1: Overview three networks: characteristics (columns 2–5); parameters values (6–9). Results: SO and UE known from literature (10–11); average travel time (and std. dev.) for GA alone (column 12), for QL alone (13), and for GA under GA<->QL approach (last column)

									Results (avg. travel time)				
			OD			GA	QL		Literature		Our Approach		
Net.	$\|V\|$	$\|E\|$	pairs	$n = \|\mathcal{A}\|$	k	m	α	δ	SO	UE	GA	QL	GA / GA<->QL
OW	24	24	4	1700	8	0.001	0.5	0.99	66.93	67.16	68.75 (.14)	67.17 (.1)	66.97 (.01)
SF	24	76	528	3606 (x100)	4	{0.01,0.001}	0.5	0.99	19.95	20.74	53.2 (1.13)	21.0 (.03)	20.83 (.02)
Braess	4	5	1	4200	3	0.01	0.5	0.99	15	20	15 (.001)	16.9 (.66)	15.02 (0.01)

As mentioned, this approach is used here to solve a TAP instance. Given a particular traffic network, the TAP seeks to assign a demand (trips, vehicles, driver agents) to links of the network. Therefore, in the TAP, a solution is a route for each agent. This can be done by computing the user equilibrium (UE), or the system optimum (SO). The UE assumes that each driver performs adaptive route choices until the agent perceives that all routes between its origin and destination (an OD pair) have minimum costs. This means that the UE is computed *individually*. On the other hand, the assignment that leads to the SO is computed *centrally* (e.g., by an optimization procedure, which is based, for instance, on the minimization of the travel time *over all* users).

For the TAP, the synergy between the central authority (computing routes for each agent) and the learning processes by the agents, as shown in Fig. 1: the central authority uses GA to compute the SO and informs agents which actions are recommended in order to achieve the SO (this is the GA → QL part); these agents periodically follow it, but mostly they learn to select their own actions (routes) by means of QL, and then inform the central about these actions (this is the QL → GA part).

Due to lack of space, details are omitted; briefly, standard procedures for GA (with elitism) and QL (with action selection based on ε-greedy) are used. Each chromosome of the population of solutions is a list indicating one route per agent. For the QL, the actions available to agents are the selection of one among k shortest routes.

In order to illustrate the use of the approach, the following traffic assignment scenarios were used: the one in [1] (OW), the Braess Paradox, and a more realistic benchmark called Sioux Falls (SF).

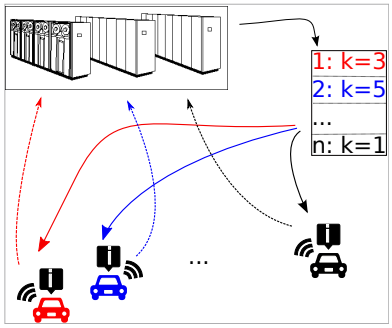

Figure 1: Synergy between central authority using a GA to compute the SO and agents learning to select routes using QL.

Their characteristics are summarized in Table 1 (columns 1–5). Please refer to the literature to see details such as cost functions for the networks.

For each network, tests were performed to determine the best values for k (number of shortest routes), m (mutation rate), c (crossover) and for the QL parameters such learning rate α and decay for ε (columns 6–9) in Table 1.

Finally, this table shows the results in terms of average travel time: from the literature (columns 10–11) and in three other situations (columns 12–14; in these cases, over 30 repetitions). First, when only GA is used to compute an approximation for the SO. Second, when only QL is used (this approximates the UE). Lastly, when both GA and QL exchange solutions. The latter requires setting the value of an extra parameter: Δ, which is the frequency with which the GA recommends solutions to the agents. In the experiments $\Delta = 10$ was used.

As shown, the SO is not achieved using GA alone, especially for the SF network. Here the GA<->QL approach is more efficient when compared to the GA alone. As for the UE, QL alone is not always able to reach it. In the Braess network, the GA<->QL not only helps agents to learn to align their actions with the global objective, but also slightly accelerates the convergence regarding the pure QL.

3 CONCLUSIONS AND FUTURE WORK

In this paper we address the issue of aligning system and user optima by means of metaheuristics and reinforcement learning respectively. To this end, a synergy between these two is proposed. We describe the use of this synergy in a particular problem related to how to assign routes to agents in a traffic network. Our results show that this synergy is able to find better solutions.

ACKNOWLEDGMENTS

Ana Bazzan is partially supported by CNPq.

REFERENCES

[1] Ana L. C. Bazzan and Camelia Chira. 2015. Hybrid Evolutionary and Reinforcement Learning Approach to Accelerate Traffic Assignment (extended abstract). In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*, R. Bordini, E. Elkind, G. Weiss, and P. Yolum (Eds.). IFAAMAS, 1723–1724. http://www.aamas2015.com/en/AAMAS_2015_USB/aamas/p1723.pdf

[2] M. Colby and K. Tumer. 2012. Shaping Fitness Functions for Coevolving Cooperative Multiagent Systems. In *Proceedings of the Eleventh International Joint Conference on Autonomous Agents and Multiagent Systems*. Valencia, Spain, 425–432.