Multi-Agent Election-Based Hyper-Heuristics

Vinicius Renan de Carvalho, Jaime Simão Sichman
University of São Paulo, São Paulo, Brazil
vrcarvalho@usp.br, jaime.sichman@usp.br

Abstract

Hyper-heuristics are high-level methodologies responsible for automatically discover how to combine elements from a low-level heuristic set in order to solve optimization problems. Agents, in turn, are autonomous component responsible for watching an environment and perform some actions according to their perceptions. Thus, agent-based techniques seem suitable for the design of hyper-heuristics. This work presents an agent-based hyper-heuristic framework for choosing the best low-level heuristic. The proposed framework performs a cooperative voting procedure, considering a set of quality indicator voters, to define which multi-objective evolutionary algorithm (MOEA) should generate more new solutions along the execution.

1 Context and Motivation

Multi-objective optimization problems (MOP) are ubiquitous in many real-world problems. In these problems, the solutions should optimize different and often conflicting criteria. Usually, classical exact optimization methods cannot be used to deal with MOPs and more sophisticated heuristic techniques are required. Multi-Objective Evolutionary Algorithms (MOEAs) have been successfully applied to solve MOPs. These algorithms are inspired by the Darwinian principles of nature's capability to evolve organisms well adapted to their environment.

Choosing an evolutionary algorithm to solve a particular optimization problem is a non-trivial undertaking. Without detailed prior information as to which particular algorithm or algorithm configurations to use for a given optimization problem, algorithm tuning is required. This may involve solving multiple problem instances using different algorithms and configurations in order to find the best one according to some quality indicator. Furthermore, due to the non-deterministic nature of evolutionary algorithms, this process may need to be repeated multiple times. Hyper-heuristics, that is *heuristics to choose heuristics*, have been proposed as a means to both simplify and improve algorithm choice and configuration for optimization problems [Burke *et al.*, 2013]. The hyper-heuristic motivation came from the *No Free Lunch Theorem*, which establishes that ”for any algorithm, any elevated performance over one class of problems is offset by diminished performance over another class” [Wolpert and Macready, 1997].

2 Research Goals and Expected Contributions

This research has been proposing an agent-based hyper-heuristic for online selecting MOEAs, that means, while the MOP is being solved, our approach finds which is the most suitable MOEA to solve it. Thus, this research aims to diminish the effort on choosing the best MOEA without performing offline training or tuning. This is an important goal due to the fact MOEAs are algorithms widely applied in several areas, and difficult reductions may help different researchers. Another goal is combining different MOEAs in order to better explore the search space, and by doing it find better solutions than using the state-of-art MOEAs.

3 Related Work

In the literature, we find several online selection hyper-heuristics in multi-objective context, usually applied to online selecting the best heuristic. Other researchers focus on selecting MOEAs. [Vázquez-Rodríguez and Petrovic, 2012] employed a set of MOEA composed by NSGA-II, SPEA2, and IBEA competing for populating the main population with their solutions. This is decided by quality indicators associated with probabilities. [Maashi *et al.*, 2014] proposed an online selection hyper-heuristic based on the Choice Function. To choose the best one, each MOEA were evaluated according to Hypervolume, RNI, UD, and AE quality indicators, a two-ranking system was performed, and the chosen MOEA was determined by a mathematical function. Their work employed NSGA-II, SPEA2, and MOGA. [Acan and Lotfi, 2016] proposed a cooperative multi-agent system of meta-heuristic agents who share the main population of solutions based on dominance ranks of its solutions. So, each multi-objective meta-heuristic is assigned to a subpopulation according to the cyclic or round-robin order. Thus, each meta-heuristic agent generates solutions using a different-rank subpopulation in subsequent. In this approach, there is the main population and all meta-heuristic agent have their own population. This work was evaluated using CEC09 benchmark.

In [de Carvalho and Sichman, 2017] we proposed MOABHH, an agent-based multi-objective online hyper-heuristic based on the concept of voting. The size of the share was determined by Copeland Voting, where a set of five quality indicators, composed by Hypervolume, Spread, RNI, GD and IGD evaluated a set of MOEA agent composed by NSGA-II, IBEA and SPEA2. After the voting, the most voted candidate received a bigger share. This work was evaluated using WFG continuous optimization benchmark, no discrete problems were employed to evaluate the proposed work, and the use of GD and IGD makes the task of adapting this work to deal with real-world problem, due to the fact of these indicators need previous problem knowledge. In [de Carvalho and Sichman, 2018] IGD and GD were replaced by HR and ER in order to solve four real-world engineering optimization problems. This paper also set GDE3 as MOEA agent.

4 Proposal

In this section, we present our hyper-heuristic for multi-objective optimization problems, which is an online selection hyper-heuristic that uses voting rules from the social choice literature to guide the choice of evolutionary algorithms. In our proposal, three kinds of agents are considered: *MOEA Agents* (Candidate), *Voter Agents* and a *HH agent* (Hyper-Heuristic Agent). Briefly speaking, our proposal has 3 main steps:

1. Given a share of the population as input, each *MOEA Agent* generates new off-spring solutions;
2. Each *Voter Agent* evaluates the output of all *MOEA Agents* and ranks these outputs (and indirectly the *MOEA Agent*) according to its evaluation criterion. Using different *Voter Agents* allows analyzing the solutions generated by distinct *MOEA Agents* under complementary perspectives;
3. The *HH agent* uses these rankings to generate a final ranking of the *MOEA agents*. The top-ranked *MOEA agent* receives a larger share of the population.

Our proposal is illustrated in Algorithm 1. First, all agents are initialized, the problem is instantiated and a random population of solutions is created. While a stopping criterion is not reached the algorithm can either uniformly split the main population or split according to voting outcomes. This is necessary because *MOEA agents* may have good performance at the beginning of the search. Thus, the task of comparing algorithm performance becomes harder. Thus, MOABHH does not perform any quality evaluation until γ generations. After that, the voting process starts by *Voter agents* evaluating MOEA related solutions and ranking preferences. After all *Voter agents* vote, *HH agent* summarises the voting preferences, according to the voting method employed. The MOEA Candidate Agent that was the election winner then receives a larger number of solutions to generate in the next generation, while the others receive a smaller share of solutions to generate. After updating the number of solutions to generate, all MOEA Candidate Agents execute over n generations and add new solutions until the next election.

Algorithm 1: MOABHH WorkFlow.

```
begin
    Initialize agents and generate a random population of solutions;
    while A stopping criterion is not reached do
        if Is It time to vote? then
            Voters evaluate MOEAs outcome and vote;
            HH agent performs the voting method;
            HH agent shares the number of solutions to generate;
        end
        else
            Uniformly share the number of solutions to generate;
        end
        MOEA agents execute for $n$ generations;
        Update the main population;
    end
end
```

5 Partial Results and Further Work

In our previous work, we had shown our approach which used Copeland voting method and GD and IGD quality indicators (which need previous Pareto knowledge) found competitive results against the best MOEA in WFG continuous framework. Then, we decided to remove GD and IGD to make our approach better suitable to real-world problems. Now we have compared Copeland against Borda and Kemeny-Young using WFG framework and the multi-objective Travel salesperson problem (moTSP), a discrete problem. Our results have shown Borda as the best voting procedure, finding competitive results in 89/90 tested problems. As further work, we intend to employ different voting methods, increase the number of MOEA candidates up to 10, and use more benchmarks and real-world problems.

References

[Acan and Lotfi, 2016] A. Acan and N. Lotfi. A multiagent, dynamic rank-driven multi-deme architecture for real-valued multiobjective optimization. *Artificial Intelligence Review*, 48:1–29, 2016.

[Burke *et al.*, 2013] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, and R. Ozcan, E.and Qu. Hyper-heuristics: a survey of the state of the art. *Journal of the Operational Research Society*, 64(12):1695–1724, Dec 2013.

[de Carvalho and Sichman, 2017] V. R. de Carvalho and J. S. Sichman. Applying Copeland Voting to Design an Agent-Based Hyper-Heuristic. In *Proc. of the 16th Conference on Autonomous Agents and MultiAgent Systems*, pages 972–980, 2017.

[de Carvalho and Sichman, 2018] V. R. de Carvalho and J. S. Sichman. Solving real-world multi-objective engineering optimization problems with an Election-Based Hyper-Heuristic. In *International Workshop on Optimisation in Multi-agent Systems (OPTMAS'18)*, 2018.

[Maashi *et al.*, 2014] M. Maashi, E. Özcan, and G. Kendall. A multi-objective hyper-heuristic based on choice function. *Expert Systems with Applications*, 41(9):4475–4493, 2014.

[Vázquez-Rodríguez and Petrovic, 2012] J. A. Vázquez-Rodríguez and S Petrovic. A mixture experiments multi-objective hyper-heuristic. *Journal of the Operational Research Society*, 64(11):1664–1675, 2012.

[Wolpert and Macready, 1997] D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Trans. on Evol. Comput.*, 1(1):67–82, Apr 1997.