New Adaptive Selection Strategies for Distributed Adaptive Metaheuristic Selection

Christopher Jankee, Sébastien Verel, Bilel Derbel, Cyril Fonlupt

New Adaptive Selection Strategies for Distributed Adaptive Metaheuristic Selection

Christopher Jankee
Université du Littoral Côte d'Opale, LISIC

Sébastien Verel
Université du Littoral Côte d'Opale, LISIC

Bilel Derbel
Université Lille 1, LIFL – CNRS – INRIA Lille

Cyril Fonlupt
Université du Littoral Côte d'Opale, LISIC

ABSTRACT

Distributed Adaptive Metaheuristics Selection (DAMS) is a framework dedicated to adaptive optimization in distributed environments. We investigate the design of adaptive strategies allowing to control the local selection of metaheuristics and to coordinate their local executions with the aim of maximizing the performance of the whole distributed system. Inspired by the multi-armed bandit framework, we propose two distributed strategies. Our experimental analysis is performed on the simple oneMax problem for which the best metaheuristics that should be executed is known.

Categories and Subject Descriptors

I.2.8 [**Computing Methodologies**]: Artificial Intelligence—*Problem Solving, Control Methods, and Search*

1. INTRODUCTION

Metaheuristics such as evolutionary algorithms are powerful methods for solving optimization problems in a black-box context when no information on the optimization function is neither available nor required. Since the early ages of evolutionary algorithms, a wide range of algorithms have been proposed, and one of the main difficult question is the choice of the "best" algorithm for a given problem to solve. Obviously, this choice should be guided by the particular features of the problem. Unfortunately, in a black-box context, the features of the problem could be hard to extract, and the choice of a relevant algorithm becomes even more difficulty.

The algorithm selection consists in selecting the best algorithm to solve a given problem. The original framework of algorithm selection has been proposed by Rice [4].

Nowadays, the off-line tuning methods can benefit from the parallel computation systems to perform a large bench of experiments in order to learn the correct parameters. In on-line control methods, the algorithm is selected along the optimization process. At each round, an algorithm is selected from a portfolio of algorithms according to their performance in previous rounds. This on-line selection problem can be model by the (dynamic) multi-arm bandit problem: each arm is an optimization algorithm, the reward is based on the quality of produced solutions by the algorithm, and the objective is to select the arms during the optimization process in order to maximize the quality of the final solution.

In this work, following the work on Distributed Adaptive Metaheuristic Selection (DAMS) [2], we investigate online portfolio methods in distributed environment where a number of computation nodes are available. The goal of this paper is to propose and analyze new adaptive selection strategies in the DAMS framework.

ACM ISBN 978-1-4503-2138-9.
DOI: http://dx.doi.org/10.1145/2739482.2764694

2. NEW ADAPTIVE SELECTION STRATEGIES FOR DAMS

We distinguish two extreme types of selection strategies according to the information sharing between computation nodes. In *independent* selection strategies, the metaheuristic selection depends only on the reward information produced by the node. In *collective* selection strategies, the selection also takes into account the reward information given by the neighboring nodes.

2.1 Independent selection strategies

The Select-Best-and-Mutate (SBM) have been proposed in [2]. The SBM strategy selects the best rewarded metaheuristics from neighboring nodes (including the node itself) encountered at last round with rate $1 - p_m$, and select one random metaheuristic with rate p_m. From this original strategy, an independent selection strategy can be designed. Instead of selecting the best rewarded metaheuristic, we can select the best rewarded metaheuristic encountered over the last W rounds by this node. In that way, no reward information from the neighbor is used. The collective original SBM strategy will be denoted as SBMc, and the new independent SBM as SBMi. The SBMi is defined with two parameters, the mutation rate p_{mut}, and the windows size W.

The Adaptive Pursuit (AP) belongs to the class of probability matching algorithms. AP is a classical adaptive selection strategy used in optimization [5], and can be used as an independent selection strategy. Several Upper Confidence Bound (UCB) algorithms have been used in the context of adaptive metaheuristic selection, see [3] for a review. The UCB strategy is an optimal strategy for stationary problem with independent arms which is not the case for adaptive

metaheuristic selection. The average empirical reward could be far from the current new reward. To overcome this drawback, the average empirical reward can be computed over a slicing windows with the last W rounds. This variant of UCB strategy is denoted UCB-Wi. At last, a dynamic version of UCB has been used in [3] that uses the Page-Hinkley test to detect whether the empirical rewards collected for the best metaheuristic has changed significantly. For the details of the test computation, please refer to page 6 of [3]. This selection strategy is denoted by UCBP-PHi, and required two parameters: the restart threshold γ and the robustness threshold δ.

2.2 Collective selection strategies

Every independent selection strategies can be used to define a collective selection strategy that takes into account the reward information sharing by the neighboring computation nodes. For instance, the empirical average of reward is estimated using the information given by the neighboring nodes. Those collective strategies versions are denoted respectively SBMc, APc, UCBc, UCB-Wc, UCB-PHc.

3. EXPERIMENTAL ANALYSIS

Following previous works [6] [2] [3] [1] on adaptive portfolio selection, we also use the well known oneMax problem, which counts the number of 1 in a bit string. In a similar scenario, we use a portfolio of four $(1 + \lambda)$-ES. The four ES only differs by the stochastic operator used: from one parent solution, the algorithm produces λ solutions according to a stochastic operator and selects the best one for the next iteration. Four operators are used: three respectively flip exactly at random 1, 3 and 5 bits, and one uniformly flips each bit with rate $1/N$ where N is the bit strings size set to $N = 1000$. We study four topologies of network: the complete topology, a random topology, the grid topology and the circle topology. The size of the networks is $n \in \{4, 16, 32, 64\}$. In order to have the same number of fitness evaluations in one round whatever the network size n, the λ parameter is set to $64/n$.

Overall Performance. From a purely distributed perspective, the first interesting measure is the number of rounds it takes for an algorithm to find the global maximum. The number of rounds provides an idea about the degree of parallelism in an ideal scenario where the communication cost is assumed to be negligible compared to the cost of function evaluation. The best performing parameters are set for each strategies. First, the performance of the different strategies are consistent with the considered configurations in the sense that they can overall be ranked similarly independently of the topology type or graph size. Interestingly, this impact of exchanging the reward information between nodes is positive in the case of SBM and AP, whereas it is not when considering UCB. In fact, SBMc appears to overall outperform all the other strategies and APc appears to performing best when both considering the circle, grid and random topologies with large number of nodes.

Adaptive Behavior. In order to better understand the properties of the different methods. The SBMc strategy, the nodes have a tendency to cluster around the best performing operator (which is the 5-bits). This is because as soon as one node has detected that this operator is actually having a good reward, the information is spread among the other nodes and the selection mechanism forces all nodes to choose this operator. In contrast, we can observe that for the APc and UCB-HPi, this is not the case since the selection mechanism at each node is less elitist compared to SBMc. Overall, the three strategies are able to deal with a reasonable tradeoff between exploitation and exploration when selecting the metaheuristic – with UCB-HPi being the strategy where the exploration is the more pronounced.

Parallelism. In the previous discussions, we were only interested in analyzing the relative behavior of the strategies for a fixed topology. It is important to recall that the number of function evaluations at every single round and for all the considered configurations is the same which means that the number of function evaluations needed overall in any of the considered configuration is by the *same* multiplicative factor similar to the number of rounds. This observation has an important impact, since then, we are able to obtain different trade-offs when considering the number of exchanged messages as an important indicator of *parallel speed-ups* that one could obtain when effectively deploying our strategies in a real distributed setting. In fact, the number of messages needed to exchange information is exactly the number of rounds times the number of edges used in the considered topology. In the case of the complete (resp, circle, grid, random) topology, the number of edges is $n(n-1)/2$ (resp. $n-1$, $O(n)$, $O(p.n^2)$) where n is the number of nodes. This is to contrast with the complete topology where the increase in the number of messages is polynomial. Hence, in a practical setting where the cost of message-passing is non-negligible, we claim that the best choice would be the random topology which exhibits the most appealing tradeoffs in terms of the number of rounds *v.s.* the number of messages exchanged overall.

4. CONCLUSION

In this work, we investigate new adaptive strategies for distributed metaheuristic selection. Accordingly, we explored the applicability of adaptive pursuit and upper bound confidence based algorithms in the distributed setting where several heterogeneous islands have to cooperate in order to select the most accurate metaheuristic dynamically at runtime. In particular, we consider the possibility of incorporating the distributed information coming from the neighboring islands and study its impact on the search behavior by considering independent and collective schemes. We find that special care must be taken when attempting to use the rewards observed distributively at different islands in order to obtain accurate exploration-exploitation trade-offs.

5. REFERENCES

[1] C. Candan, A. Goeffon, F. Lardeux, and F. Saubion. A dynamic island model for adaptive operator selection. In *GECCO '12*, pages 1253–1260, 2012.

[2] B. Derbel and S. Verel. DAMS: distributed adaptive metaheuristic selection. In *GECCO '11*, pages 1955–1962, 2011.

[3] A. Fialho, L. Da Costa, M. Schoenauer, and M. Sebag. Analyzing bandit-based adaptive operator selection mechanisms. *AMAI*, 60:25–64, 2010.

[4] J. R. Rice. The algorithm selection problem. *Advances in Computers*, 15:65–118, 1976.

[5] D. Thierens. An adaptive pursuit strategy for allocating operator probabilities. In *GECCO'05*, pages 1539–1546,

2005.

[6] S. Tongchim and P. Chongstitvatana. Parallel genetic algorithm with parameter adaptation. *Information Processing Letters*, 82(1):47 – 54, 2002.