SHORT TERM SCIENTIFIC MISSION (STSM) – SCIENTIFIC REPORT

The STSM applicant submits this report for approval to the STSM coordinator

Action number: CA15140
STSM title: Multi-Dimensional Parameter Optimization for the (1+(lambda,lambda)) GA with irace
STSM start and end date: 19/09/2018 to 26/09/2018
Grantee name: Nguyen Dang

PURPOSE OF THE STSM/

The (1+(λ,λ)) GA developed in [Doerr, Doerr and Ebel, 2015] is known to optimize OneMax more efficiently than any unary unbiased (i.e., mutation-based) algorithm. More precisely, it is known that its average optimization time can be as small as linear, when the parameters of the algorithm are suitably chosen. This was shown in [Doerr and Doerr, 2018]. Apart from these theoretical considerations, however, we do not yet have gained a solid understanding of how to choose the multiple parameters of the (1+(λ,λ)) GA to obtain best possible running time on realistic problem dimensions. Automated parameter tuning tools such as irace [López-Ibáñez et al, 2016] are designed to support the user in selecting suitable parameter values. The objective of this STSM is to use such tuning tools to optimize the performance of the (1+(λ,λ)) GA. After the tuning step, post-analysis is applied to gain more understandings on how sensitive the parameters are on the performance of the algorithm.

DESCRIPTION OF WORK CARRIED OUT DURING THE STSMS

We first study parameters of the static version of the algorithm. These include four parameters: the mutation rate $p = k/n$, where n is problem size, the offspring sizes λ_1 (mutation phase) and λ_2 (crossover phase), and the crossover bias *c*. *irace* is used to tune those parameters on each problem size *n* in {500,1000,1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000}. At first, the ranges of those parameters are set to be large, as we were not certain about which values should give good performance. After the first tuning and a post analysis on the parameter sensitivity using the tool fANOVA [Hutter et al, 2011], we were able to identify the most important parameter, λ_1, and its corresponding range of good values. We then proceed on doing a second tuning where λ_1 is limited to the good range. The aim is to further investigate the roles of other parameters in the local region of parameter space where algorithm performance is reasonably good. Results of this first study are used as a baseline for the next two studies described in the following paragraphs.

The second study is on parameter control of the same algorithm. Mutation rate, offspring sizes and crossover bias are no longer fixed. They are updated during the search using some factors. In particular, we set $\lambda_1 = \lambda_2 = \lambda$, $p = \lambda/n$ and $c = 1/\lambda$. The initial value for λ is 1. When an iteration is successful and a better solution is found, λ is decreased by a factor of *b*, and is increased by a factor of *a* otherwise. We then use irace to tune a and b.

In the third study, we go one step further by exposing more parameters to the parameter control strategy. More specifically, we set $p = \alpha * k / n$, $\lambda_1 = \lambda$, $\lambda_2 = \beta * \lambda$ and $c = \gamma / \lambda$. We then tune the five parameters α, β, γ, a and b using irace. Further post-analyses are then done to gain more insights into the influence of the hyper-parameters on the performance of the algorithms.

DESCRIPTION OF THE MAIN RESULTS OBTAINED

Results of the first study gives some interesting insights into the algorithm parameters: the most important parameter is the mutation offspring size λ_1. On the problem sizes tested, it is crucial to set this parameter sufficiently small (within the range of 4 and 10) in order to obtain good performance. Another observation is that the parameter setting suggested in the theoretical work [Doerr, Doerr, Ebel 2015] is able to obtain similar performance as the ones found by irace.

Results of the second study suggest parameter settings that are close to the 1/5th rule. This suggestion is consistent across different problem sizes. This is an interesting result and indicate the potential of combining theory and experimental works. Performance obtained from the tuning in this study is significantly better than the ones found in the static setting, which confirms the advantage of parameter control over fixed parameter settings.

In the third study, irace is able to find parameter setting that statistically significantly improves performance of the ones found in the second study. Results indicate consistency on the best values for all five parameters, and we were able to derive a configuration that perform well on all tested problem sizes. This configuration offers 15% reduction of the average running time compared to the dynamic version with two hyper-parameters. The stable performance of the tuned configurations indicates that a precise running time analysis might be possible.

The work in this STSM results in the following publication:

Dang, N. & Doerr, C. (2019) **Hyper-parameter tuning for the (1+ (λ, λ)) GA**. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '19). New York: ACM, p. 889-897

FUTURE COLLABORATIONS (if applicable)

We are considering extending the studies to more practical problems, such as MaxSAT. Another possible future collaboration is on using grammatical evolution to automatically discover the best algorithm control strategies for the dynamic version of (1+(λ,λ)) GA.

REFERENCES

Doerr, B., Doerr, C., & Ebel, F. (2015). From black-box complexity to designing new genetic algorithms. *Theoretical Computer Science*, 567, 87-104.

López-Ibáñez, M., Dubois-Lacoste, J., Cáceres, L. P., Birattari, M., & Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3, 43-58.

Hutter, F., Hoos, H., & Leyton-Brown, K. (2014). An efficient approach for assessing hyperparameter importance. In *International Conference on Machine Learning* (pp. 754-762).