‘Hang On a Minute’: Investigations on the Effects of Delayed Objective Functions in Multiobjective Optimization

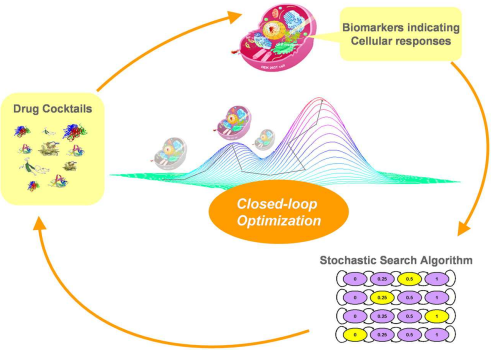
Richard Allmendinger [1] and Joshua Knowles[2]

[1]University College London
[2]University of Manchester

March 20, 2013

Closed-loop optimization

Applications include: shape design optimization, experimental quantum control, drug discovery, instrument optimization, taste optimization, ...



[image from PK Wong (2008), *PNAS* **105**(13)]

Delayed objective functions

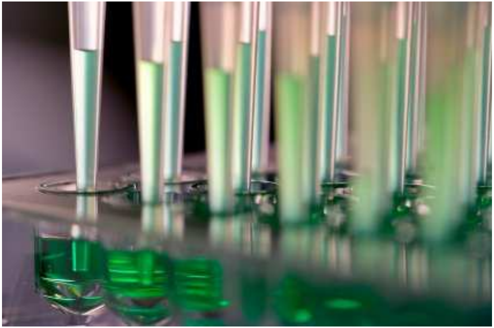
Batch evaluation

Assumption: experiments are done in batches

Delayed objective functions

Batch evaluation

Assumption: experiments are done in batches

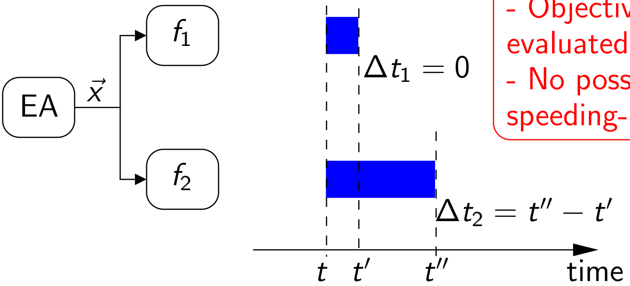


Delayed objective functions

Focus of research

Multiobjective optimization problems where at least one of the objective functions requires a relatively longer time to be evaluated than the cheapest/quickest of the objective functions → at any given time, fitness estimates of some solutions may only be partial

Δt_i - Evaluation delay of objective i relative to the quickest objective

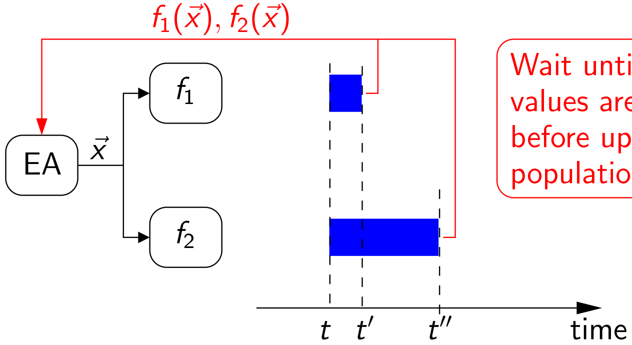


- Objectives are evaluated in batches
- No possibility of speeding-up evaluations

Delayed objective functions

Focus of research

Multiobjective optimization problems where at least one of the objective functions requires a relatively longer time to be evaluated than the cheapest/quickest of the objective functions → at any given time, fitness estimates of some solutions may only be partial

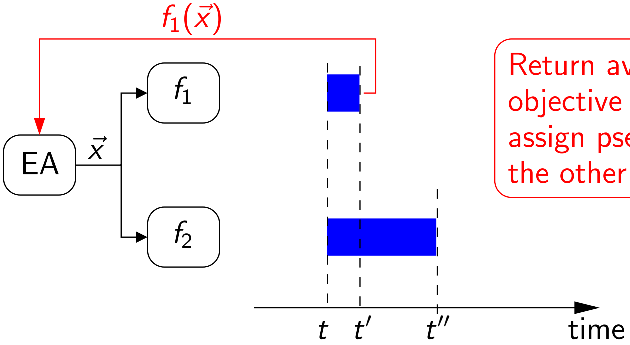


Wait until all objective values are available before updating the population

Delayed objective functions

Focus of research

Multiobjective optimization problems where at least one of the objective functions requires a relatively longer time to be evaluated than the cheapest/quickest of the objective functions → at any given time, fitness estimates of some solutions may only be partial



Return available objective value and assign pseudovalue to the other objectives

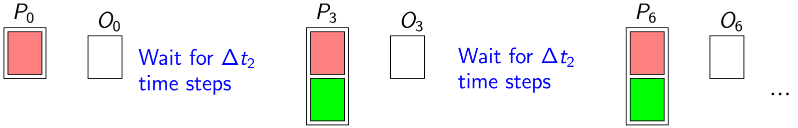
Related work

- Finding minimal sets of objective functions without conflicting with the full set (Brockhoff and Zitzler, 2009)
- Asynchronous evaluation in optimization in the context of grid computing (Scriven et al., 2008; Lewis et al., 2009)
- Age-layered populations to allow solutions from previous generations to take part in reproduction (Hornby, 2006)
- Estimating objective values using surrogate modeling techniques including fitness inheritance (Smith et al., 1995; Runarsson, 2004)
- Ephemeral resource constraints (Allmendinger and Knowles, 2011, 2011a, 2012; Allmendinger, 2012): Temporary limitations in the capacity to evaluate certain otherwise feasible solutions during the optimization process.

Population update strategies

- Waiting strategy: Wait until all evaluations have been completed $\rightarrow$ standard EAs and population update rules can be applied
- Non-waiting strategy: Solutions with complete and partial information on objective values co-exist in a population growing without bound

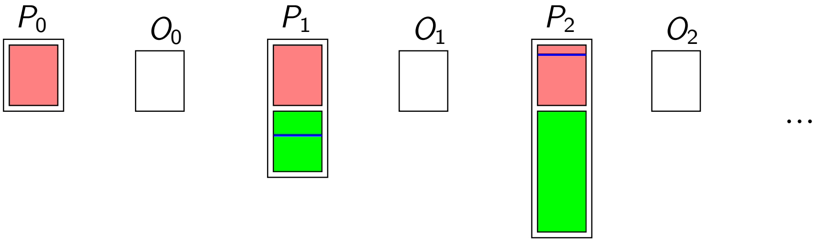
e.g. f_1 needs 1 time step to be evaluated, and f_2 has an evaluation delay of $\Delta t_2 = 2$



P_i - (Ranked) population at time step i
O_i - Offspring population at time step i

Population update strategies

- Waiting strategy: Wait until all evaluations have been completed $\rightarrow$ standard EAs and population update rules can be applied
- Non-waiting strategy: Solutions with complete and partial information on objective values co-exist in a population growing without bound

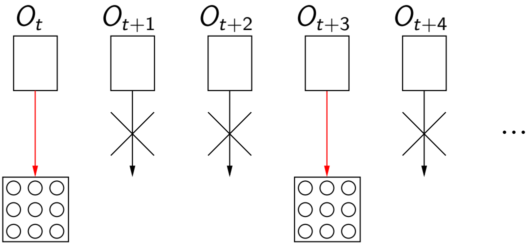


P_i - (Ranked) population at time step i
O_i - Offspring population at time step i

Selecting solutions for evaluation on the delayed objective function f_m

- Sweep selection: Select always the most recently generated solutions
- Priority-based selection: Select solutions based on a score indicating a solution's potential to change the ranking of all (completely evaluated) solutions in P

e.g. expensive objective function needs 3 time steps to be evaluated

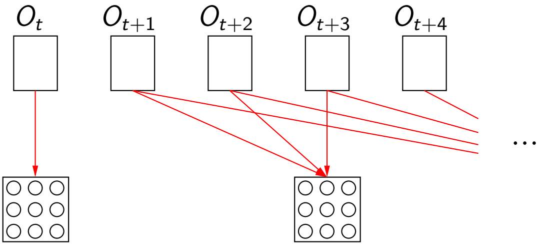


O_i - Offspring population at time step i

Selecting solutions for evaluation on the delayed objective function f_m

- Sweep selection: Select always the most recently generated solutions
- Priority-based selection: Select solutions based on a score indicating a solution's potential to change the ranking of all (completely evaluated) solutions in P

e.g. expensive objective function needs 3 time steps to be evaluated



O_i - Offspring population at time step i

Strategies for dealing with delayed objectives

Assignment of pseudovalues to the delayed objective f_m

1. Random pseudovalue assignment: Uniform variate within the observed objective range(s) um value of objective f_m of all solutions in P that have actually been evaluated on objective f_m
2. Noise-based pseudovalue assignment: Add noise to a value drawn from an existing solution value of the delayed objective
3. Fitness inheritance-based pseudovalue assignment: Use simple 1-NN scheme in decision space n all objectives

Pseudovalues are reassigned at each generation

Strategies for dealing with delayed objectives

Ranking of solutions

1. Performance ranking: Sort all solutions in P according to their non-dominated sorting ranks only
2. Performance + age ranking: Sort P based on the age of solutions where more recently generated solutions are favoured in environmental selection. Parental selection is then done on non-dominated sorting ranks of solutions.

Experimental Setup

EA parameter settings

- Ranking-based EMOA with a non-fixed population size
- For environmental selection the setting was $\mu = \lambda = 50$
- Solutions are evaluated in a batch of size $k_i = \mu, i = 1, \ldots, m$
- Binary Tournament selection, simulated binary crossover ($p_c = 0.9$), and polynomial mutation ($p_m = 1/l$)

Test problems

- WFG1-WFG9 of the Walking Fish Group toolkit (Huband et al., 2006)
- Problems consisted of $l = 6$ continuous decision variables and $m = 2$ or 3 objectives
- Evaluation delay Δt_i measured in time steps (here generations)
- 20 independent algorithmic runs were performed for each experiment

Results - Standard EA / Waiting

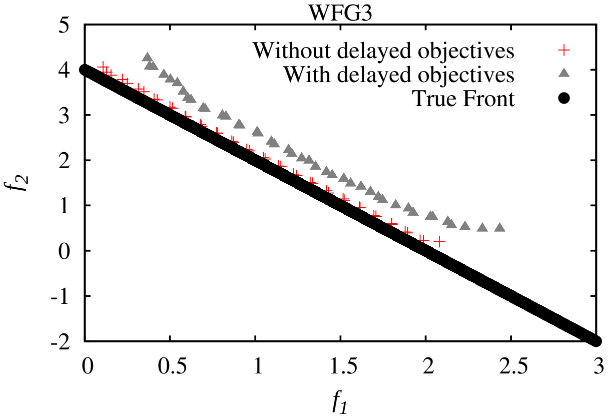
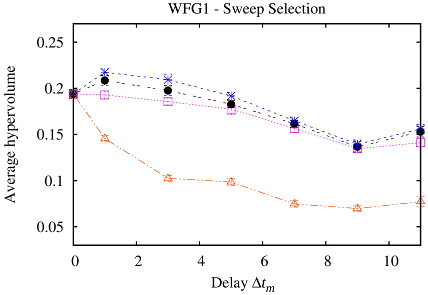


Figure : Estimated true Pareto Front and median attainment surface obtained on WFG3 with $m = 2$ objectives with objective f_2 having an evaluation delay of $\Delta t_2 = 3$ time steps. The EMOA employed a waiting strategy.

Results



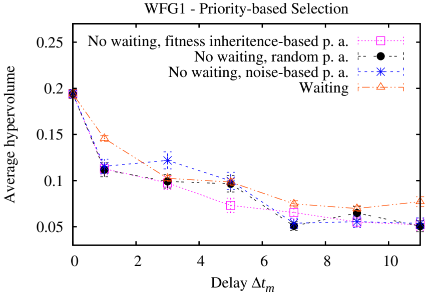


Figure : Average hypervolume on WFG1 with $m = 3$ objectives and one objective function, f_3, delayed by Δt_3.

Results - Sweep versus Priority-Based Selection

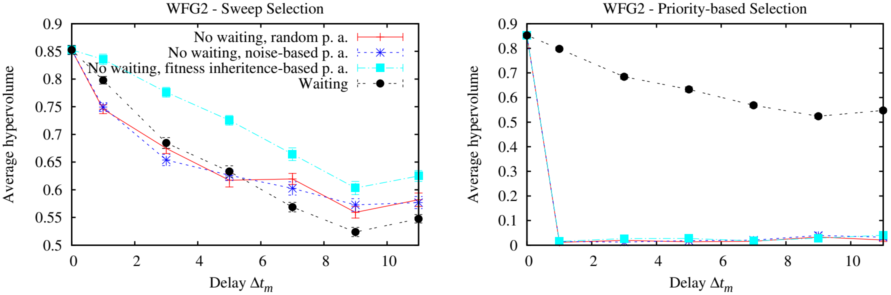


Figure : Average hypervolume on WFG2 with $m = 3$ objectives and one objective function, f_3, delayed by Δt_3.

Results - Sweep on 2- and 3-Objective Problems

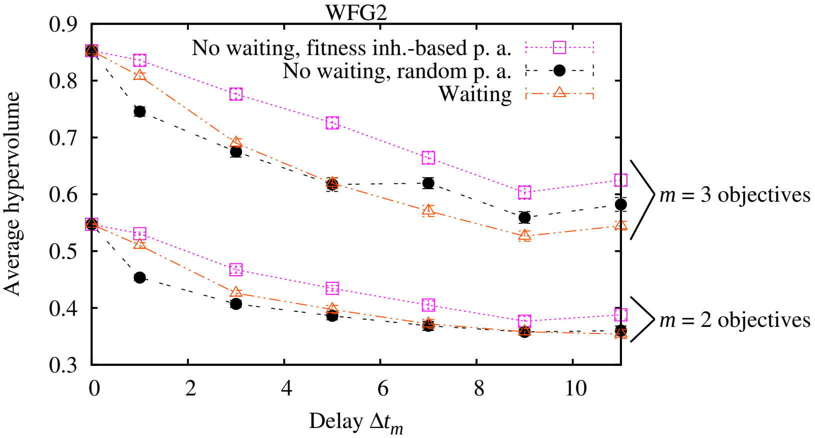


Figure : Average hypervolume on WFG2 with $m = 2$ and 3 objectives using 1 delayed objective function, f_2, with Δt_2 time steps. Sweep Selection.

Results - Sweep Selection (2 Delayed Objectives)

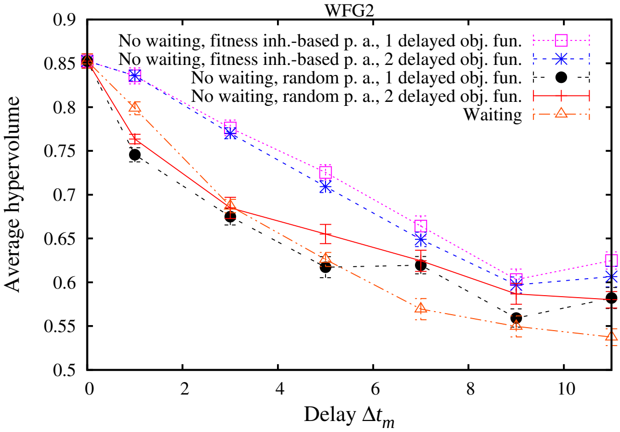


Figure : Average hypervolume on WFG2 with $m = 3$ objectives using 1 and 2 delayed objective functions, f_2 and f_3, with $\Delta t_2 = \Delta t_3$ time steps. The EMOAs employed Sweep Selection.

Conclusion

- Delayed objective functions degrade performance of a standard EA
- For short delays, waiting performs relatively well
- For longer delays:
  - employ a fitness inheritance-based pseudovalue assignment,
  - rank solutions based on performance only
  - evaluate most recently generated solutions on delayed objectives
- Observations hold on WFG2–9, for 2 or 3 objectives.

Future Work

- Improve pseudovalue assignment and selection of solutions for evaluation on delayed objectives
- Develop strategies for switching between waiting and not waiting during the optimization (Allmendinger and Knowles, 2011)
- Consider many-objective problems where several objectives are subject to delays of different durations
- Establish a framework for describing algorithms that can cope with delayed objective functions

Questions ?

Acknowledgments:
Doug Kell and many others (closed-loop applications work)
Ian Stott and Jane Shaw (Unilever)

References

R. Allmendinger. **Tuning Evolutionary Search for Closed-Loop Optimization**. PhD thesis, School of Computer Science, The University of Manchester, 2012.

R. Allmendinger and J. Knowles. **On-line purchasing strategies for an evolutionary algorithm performing resource-constrained optimization**. In Proceedings of PPSN XI, pages 161-170, 2011.

R. Allmendinger and J. Knowles. **Policy learning in resource-constrained optimization**. In Proceedings of GECCO, pages 1971-1978, 2011.

R. Allmendinger and J. Knowles. **On handling ephemeral resource constraints in evolutionary search**. Evolutionary Computation, 2013. Posted Online November 19, 2012. (doi:10.1162/EVCO a 00097).

D. Brockhoff and E. Zitzler. **Objective reduction in evolutionary multiobjective optimization: theory and applications**. Evolutionary Computation, 17(2):135-166, 2009.

G. Hornby. **ALPS: the age-layered population structure for reducing the problem of premature convergence**. In Proceedings of GECCO, pages 815-822, 2006.

A. Lewis, S. Mostaghim, and I. Scriven. **Asynchronous multi-objective optimisation in unreliable distributed environments**. Biologically-Inspired Optimisation Methods, pages 51-78, 2009.

References

T. P. Runarsson. **Constrained evolutionary optimization by approximate ranking and surrogate models**. In Proceedings of PPSN VIII, pages 401-410, 2004.

I. Scriven, D. Ireland, A. Lewis, S. Mostaghim, and J. Branke. **Asynchronous multiple objective particle swarm optimisation in unreliable distributed environments**. In IEEE Congress on Evolutionary Computation, pages 2481-2486, 2008.

R. Smith, B. Dike, and S. Stegmann. **Fitness inheritance in genetic algorithms**. In Proceedings of the ACM symposium on Applied computing, pages 345-350, 1995.

S. Huband, P. Hingston, L. Barone, and L. While. **A review of multiobjective test problems and a scalable test problem toolkit**. IEEE Transactions on Evolutionary Computation, 10(5):477-506, 2006.