Exploring the Performance of an Evolutionary Algorithm for Greenhouse Control

Rasmus K. Ursem
EVALife Research Group
Dept. of Computer Science
University of Aarhus
Bldg. 540, Ny Munkegade
DK-8000 Aarhus C, Denmark
ursem@daimi.au.dk

Bogdan Filipič
Dept. of Intelligent Systems
Jožef Stefan Institute
Jamova 39
SI-1000 Ljubljana, Slovenia
bogdan.filipic@ijs.si

Thiemo Krink
EVALife Research Group
Dept. of Computer Science
University of Aarhus
Bldg. 540, Ny Munkegade
DK-8000 Aarhus C, Denmark
krink@daimi.au.dk

Abstract. *Evolutionary algorithms for optimization of dynamic problems have recently received increasing attention. Online control is a particularly interesting class of dynamic problems, because of the interactions between the controller and the controlled system. In this paper, we report experimental results on two aspects of the direct control strategy in relation to a crop-producing greenhouse. In the first set of experiments, we investigated how to balance the available computation time between population size and generations. The second experiments were on different control horizons, and showed the importance of this aspect for direct control. Finally, we discuss the results in the wider context of dynamic optimization.*

Keywords. Evolutionary algorithms, direct control, dynamic systems, crop-producing greenhouse

1. Introduction

Optimization problems from the real world are often characterized by constraints, multiple objectives, and dynamic properties. In particular, control problems are typically dynamic because of the interaction between the controller and the controlled system. Furthermore, such problems usually contain time-varying components; for instance, materials with temperature dependent properties. This dynamic behavior poses an extra challenge to the optimization algorithm, because it must be able to cope with the changing problem. Evolutionary computation is a promising approach to dynamic optimization problems, since multiple solutions are kept in the population. Hence, the population is likely to contain a good solution to the problem after a change. Evolutionary algorithms (EAs) for optimization of dynamic problems have been studied over the past 15 years. Several algorithms have been suggested and tested, though mainly on artificial benchmark problems (for a survey see [1]). A well-investigated type of artificial dynamic problem is the so-called numerical problems, where the objective is to optimize a vector of real-valued numbers under the changing fitness landscape. A typical artificial problem consists of a few peaks that change position, height, and width at certain intervals. These artificial problems were recently scrutinized and found to have little in common with realistic dynamic problems, in particular with control problems [10].

Real control problems are usually handled by either an offline design process or an online control strategy. Tuning a PID controller is a typical example of an offline problem. Here, the EA uses a simulator to determine the best parameters for the controller, which is later installed in the real system. EAs have successfully been applied to PID controller tuning on several occasions, e.g., [5], [3]. In online control, the simulator is repeatedly used to determine the best control signals during the control period. Naturally, this approach is heavily dependent on the computation time of the simulator, and the rate at which control signals must be provided. Hence, the approach is only feasible for rather slowly changing problems where the signal calculation is allowed to take several seconds or even minutes. An example is greenhouse control, where the settings for heating, ventilation, CO_2, and water injection are updated every 15 minutes.

In this paper we focus on two aspects of online greenhouse control with EAs. This study is a follow-up investigation of the work presented in [6]. In the previous study we explored trade-offs between population size and number of generations between problem updates. Furthermore, we investigated two fitness functions and compared two setups for total number of evaluations. These investigations were carried out using a rather simple greenhouse simulator that did not model important aspects such as wind cooling, energy loss through the ground, and steam density. In this study, we have vastly improved the greenhouse simulator to include these aspects and several others [9]. To examine the new simulator, we extended our investigations regarding trade-offs between population size and number of generations between problem changes. The new setup includes a more extreme setting and a near-optimal solution. Additionally, we investigated the role of the control horizon length, which is the number of simulated time-steps used in the determination of the control signals. Naturally, the control horizon influences the computation time of the simulator. However, it may also influence the control performance, because the prediction precision decreases with longer look-ahead.

The paper is organized as follows: Section 2 explains the fundamental concepts of direct control with EAs. In Section 3, we describe the greenhouse simulator. The experimental setup and results are covered in Section 4. Finally, Section 5 contains a discussion of the results and general conclusions from this study.

2. Direct control with EAs

A control problem is often modeled by the interactions between the controller, the system, and the surrounding environment (see Fig. 1). Here, vector $\mathbf{x}(t)$ represents the internal state of the system at time t, $\mathbf{v}(t)$ is the environment state, $\mathbf{u}(t)$ denotes the control signal, and $\mathbf{y}(t)$ is the output from the system.

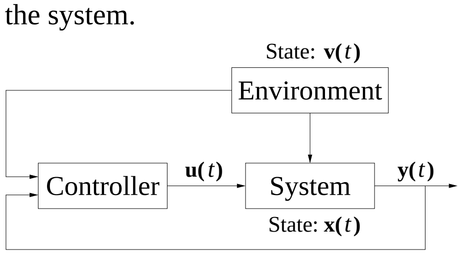

Figure 1. Model for controller, system, and environment

The change in system state is usually modeled by a number of difference equations of the form:

$$x_i(t+h) = x_i(t) + \Delta x_i(\mathbf{u}, \mathbf{x}, \mathbf{v}, t, h) \qquad (1)$$

where x_i is the i-th system variable in $\mathbf{x}$, $\Delta x_i(\cdot)$ is the update function, t is the time, h is the length of a time-step, and $\mathbf{u}$, $\mathbf{x}$, and $\mathbf{v}$ are the control signals, the system state, and the environment state of previous time-steps (sometimes several steps in the past). Real systems are often described by a set of non-linear differential equations. In these cases, an approximation method, such as Runge-Kutta, is used as the update function $\Delta x_i(\cdot)$.

The online control strategy used here is called "direct control" [4], in which the population encodes the real-valued control signals[1]. As mentioned in the introduction, the control signals must be updated at certain intervals. Hence, only a limited number of evaluations is possible between updates of the control signals. However, the number of evaluations ($\#ev$) can be balanced between population size (ps) and number of generations (gen), i.e., $\#ev = ps \cdot gen$. For instance, 200 evaluations can be assigned as either $ps = 200$, $gen = 1$ or $ps = 25$, $gen = 8$. Another important aspect of direct control is the control horizon (*CH*), which is used in the evaluation of candidate solutions. The fitness of a solution is determined by its control performance for *CH* time-steps into the future. The best control setting is then used to control the real system for one time-step. In pseudocode, the direct control algorithm is as follows:

```
Direct control
Initialize population of size ps
while(control period not over) {
Reset best control setting
   for (i=0; i < gen; i++ ) {
      Crossover and mutation
      Evaluate each solution for CH steps
      Selection
      Store best control setting
   }
   Let best setting control one step
}
```

Direct control shares many properties with the control engineering approach known as generalized predictive control (GPC) [2]. However, GPC is not easily applied to non-linear problems, because the determination of the control signals, in this case, relies on minimization of a multimodal function, which is generally not possible with traditional engineering techniques.

[1] In [4], the technique is called "direct optimal control"; however, "optimal" is a bit misleading.

3. The greenhouse control problem

The crop-producing greenhouse is modeled as illustrated in Fig. 1. The control, system, and environment variables are listed in Table 1.

Table 1. Control, system, and environment variables in the greenhouse

	Description	Var.
Control	Heating [W/m^2]	u_{heat}
	Ventilation [m^3/(m$^2 \cdot$ h)]	u_{vent}
	CO_2 injection [g/(m$^2 \cdot$ h)]	u_{CO2}
	Water injection [g/(m$^2 \cdot$ h)]	u_{water}
System	Indoor steam density [g/m^3]	x_{steam}
	Indoor air temperature [°C]	x_{atemp}
	Indoor CO_2 concentration [ppm]	x_{CO2}
	Accumulated biomass [g/m^2]	x_{biom}
	Accumulated profit [DKK/m^2]	x_{profit}
	Condensation on glass [g/m^2]	x_{cond}
Environment	Outdoor sunlight intensity [W/m^2]	v_{sun}
	Outdoor air temperature [°C]	v_{atemp}
	Outdoor ground temperature [°C]	v_{gtemp}
	Relative humidity [% r.H.]	v_{RH}
	Wind speed [m/s]	v_{wind}
	Outdoor CO_2 concentration [ppm]	v_{CO2}
	Price of heating [DKK/(W·h)]	v_{Pheat}
	Price of CO_2 [DKK/kg]	v_{PCO2}
	Price of tomatoes [DKK/kg]	v_{Ptom}

The greenhouse is controlled by four variables for heating (u_{heat}), ventilation (u_{vent}), injection of artificial CO_2 (u_{CO2}), and injection of water (u_{water}). The range of these control variables are as follows: $u_{heat} \in [0, 150]$, $u_{vent} \in [0, 100]$, $u_{CO2} \in [0, 10]$, and $u_{water} \in [0, 100]$. The change in greenhouse state is modeled by six non-linear differential equations. The simulator is based on a German description [7]. Unfortunately, the greenhouse simulator is too complex to describe in this paper, but a complete specification is available in English in [9].

The fitness of a solution s at time t is calculated as the profit achieved minus a penalty p:

$$Fit(s,t) = \sum_{j=t}^{t+CH} \Delta x_{profit}(j) - p(j) \qquad (2)$$

where

$$p(j) = \begin{cases} 10 \cdot (16 - x_{atemp}(j)) & x_{atemp}(j) < 16 \\ 10 \cdot (x_{atemp}(j) - 35) & x_{atemp}(j) > 35 \\ 0 & \text{otherwise} \end{cases}$$

The profit is equal to the income from the produced crops minus the expenses to heating and CO_2 (Eq. 29 in [9]). The penalty is enforced to avoid damage to the crops and to ensure that the indoor air temperature is kept in the optimal range for growth.

Real weather data from the Aarslev measuring station on the island Fyn, Denmark, was used for the environment variables sunlight intensity (v_{sun}), outdoor air temperature (v_{atemp}), outdoor ground temperature (v_{gtemp}), relative humidity (v_{RH}), and wind speed (v_{wind}). The remaining environemnt variables were kept constant to $v_{CO2} = 340$, $v_{Pheat} = 0.0002$, $v_{PCO2} = 4.0$, and $v_{Ptom} = 12.0$. The weather data can be obtained for a small fee from the Danish Meteorologic Institute; see [9] for further information. In this study, we simulated the first week of May. The weather data is illustrated in Fig. 2. As stated earlier, direct control determines a solution's performance by simulating a number of steps into the future. In practice, this includes simulating the weather, or rather, predicting the weather in the control horizon. Weather prediction is generally difficult; however, a simple scheme is to assume the weather to be fixed during the control horizon (few hours).

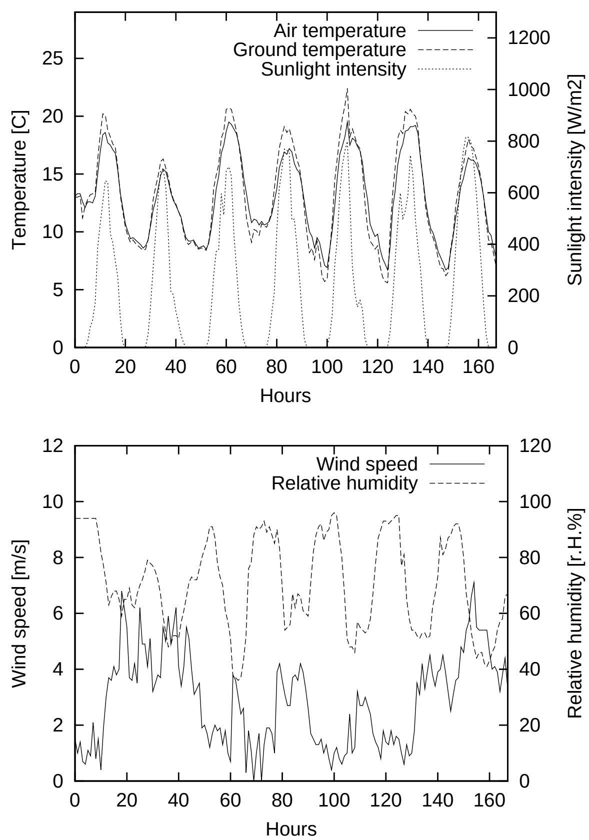

Figure 2. Weather data for the first week of May

4. Experiments and results

The simple EA encoded the four control signals as a real-valued vector. New solutions were created using Gaussian mutation and a variant of arithmetic crossover with one weight per variable. All weights except one were randomly assigned 0 or 1, and the remaining weight was set to a random value between 0 and 1. Binary tournament selection was applied. The algorithm used the following parameters: probability of crossover $p_c = 0.9$, probability of mutation $p_m = 0.5$, and variance $\sigma = 0.01$, which was scaled by the length of each control variable's interval. Each solution was evaluated by simulating *CH* time-steps using the control setting encoded in the genome. The profit achieved in each step was recorded and used to calculate the fitness (Eq. 2).

Two sets of experiments were conducted. First, we investigated five trade-offs between population size and number of generations. The trade-offs (ps, gen) were (200, 1), (100, 2), (50, 4), (25, 8), and (10, 20). Second, we tested six control horizons (*CH*) of 1, 2, 3, 4, 8, and 20 time-steps. Each experiment was repeated 30 times.

Fig. 3 illustrates the profit per m^2 in DKK for the five trade-offs using $CH = 4$. The graphs clearly show that the trade-off (10, 20) is the best. Furthermore, the order of the trade-offs shows an evident relationship between performance and number of generations – few generations lead to low performance. Hence, the available evaluations are best utilized with a low population size and many generations between problem updates. In addition to the mentioned trade-offs, we obtained a near-optimal solution solution using 10000 evaluations with $ps = 50$ and $gen = 200$. The (10,20)-trade-off was, in fact, very close to the near-optimal solution.

The control signals u_{heat} and u_{CO2} for the best setting (10, 20) and the worst (200, 1) are displayed in Fig. 4. The difference in performance is closely related to these variables, because profit is easily lost on sub-optimal control of heating and CO_2 injection. At night the temperatures drop, which requires heating to avoid damage to the crops. At daytime the sunlight permits growth, which can be augmented by injection of additional CO_2. The best control strategy (Fig. 4, upper graph) properly adjusted the control to follow the day and night phases. The worst strategy failed to turn off heat at daytime, and valuable CO_2 was wasted during the night where no growth was possible because of the absent sunlight.

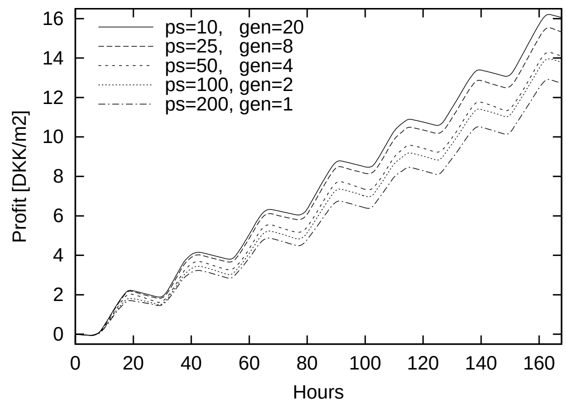

Figure 3. Population size vs. generations. Control horizon of 4 steps. Average of 30 runs

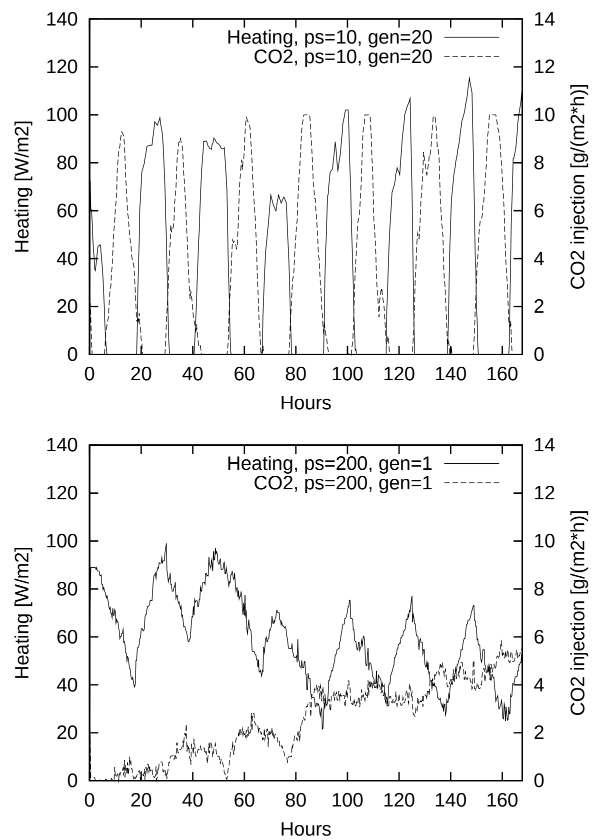

Figure 4. Example of heating and CO_2 injection for best control (upper graph) and worst control (lower graph) with a control horizon of 4 steps

In the second set of experiments, we investigated the effect of varying the control horizon. We tested six horizons having 1, 2, 3, 4, 8, and 20 time-steps. Fig. 5 shows the results from the 1, 2, 4, and 20 horizons using the (10, 20)-trade-off (to keep the graph readable, 3 and 8 are not shown). A control horizon of 20 time-steps is the best, though only marginally better than a horizon of 8 steps. The profit achieved in the remaining four horizon decreases according to the look-ahead. Hence, a control horizon of at least 8 steps yields high profit, a horizon of 4 steps leads to a reasonable good profit, and only a few steps give rather low profit. The explanation for the significant difference between the worst performing setting ($CH = 1$) and the best setting ($CH = 20$) is found by examining the control signals. Fig. 6 displays ventilation (u_{vent}) and CO_2 injection (u_{CO2}) for $CH = 1$ and $CH = 20$. The graph on ventilation shows that two general control strategies exist. The first strategy is used when $CH = 1$. Here, the EA sets ventilation high at daytime. This will require large investment in heating during night, but will utilize the free CO_2 in the environment better. The second strategy appears when $CH = 20$. In this strategy, ventilation is low, which saves some heating, but makes CO_2 injection necessary. The additional profit achieved by the 20-step controller is mainly related to the achieved temperature and indoor CO_2 level (Eq. 25 in [9]), which can be seen by thoroughly examinating the control signals and greenhouse states of both settings. Naturally, the two different control strategies emerge as a result of the control horizon, but here the CO_2 level plays an important role too. The second strategy appears because the long look-ahead allows the controller to discover the long-term effect of growth, i.e., that the photosynthesis can transforms more CO_2 than achievable by ventilation alone. Hence, additional growth is possible by injecting artificial CO_2. The short look-ahead of the first strategy does not allow the controller to discover the long-term effects. Thus, ventilation is used because it will provide free CO_2 from the environment.

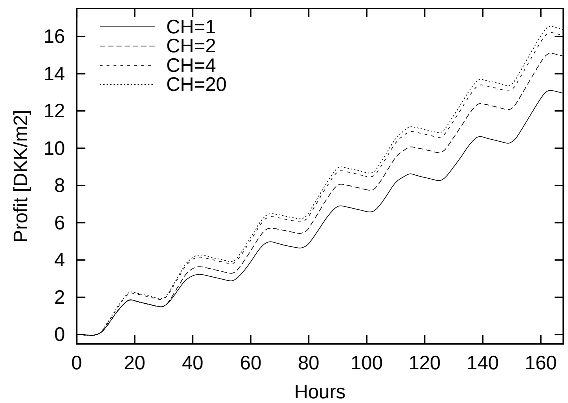

Figure 5. Profit per m^2 for different control horizons with 10 individuals and 20 generations. Average of 30 runs

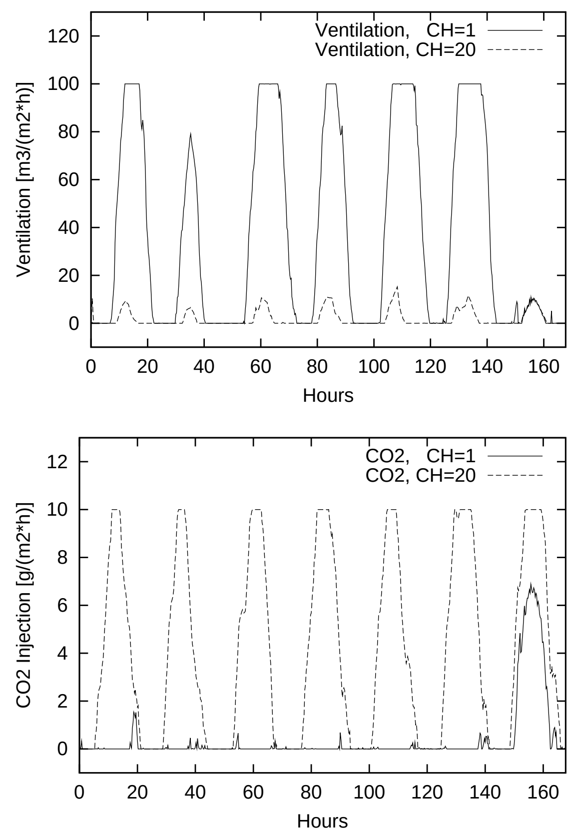

Figure 6. Example of ventilation and CO_2 injection for different control horizons (10 individuals, 20 generations)

5. Discussion and conclusions

In this paper, we investigated two important aspects of direct control with evolutionary algorithms. Our experiments show that the available number of evaluations is best invested using a low population size and many generations between updates of the problem. Applying this result essentially turns the dynamic problem into a series of related static problems. For instance, the combination of 10 individuals and 20 generations performs significantly better than a population of 25 individuals and 8 generations. This is surprising because a population of 10 individuals is generally considered to be insufficient for most static problems. Interestingly, a long static period (20-50 generations) between problem updates has been the preferred setting in most investigations on artificial dynamic problems. Furthermore, this observation confirms the preliminary results carried out in an earlier investigation conducted by the first author of this paper [8]. Regarding future work, the general trend in the trade-off experiments suggests to test even more extreme settings. In this context, a comparison with other optimization techniques, such as particle swarm optimization and simulated annealing, may be in place.

Our second series of experiments underlined the importance of choosing an appropriate control horizon. Interestingly, two very different control strategies emerged. The first strategy settled on high ventilation, much heating, low CO_2 injection, and low water injection. This strategy occurred when the control horizon was short (one time-step). The second strategy was nearly the opposite, i.e., low ventilation, medium heating, high CO_2 injection, and high water injection. This rather surprising difference is related to the long-term effects of growth, such as the CO_2-consumption by the plants. A control horizon of only one step does not reveal this, because the indoor CO_2 level does not drop drastically from one step to the next. Hence, high ventilation gives free CO_2 from the environment, which is cheaper than augmenting it artificially. An interesting question to ask in this context is would it be possible to switch to the better second strategy during the day? A possible answer is that the control setting corresponding to strategy two is more or less the opposite of strategy one. Hence, a switch would require a "jump" from one end of the search space to the other. Furthermore, following strategy one for a number of steps may actually render strategy two less profitable, because the *search itself changes the problem*; hence, the greenhouse state could be different. In a theoretical EA-context, this suggests multiple optima in *time* rather than in space. Further analysis of the greenhouse state and control traces may shed some light on these matters.

Acknowledgements

This work was supported by the Slovenian Ministry of Education, Science and Sport (project "Evolutionary Optimization of Dynamic Systems") and the Danish Research Council (EVALife project).

References

[1] Branke J. Evolutionary Approaches to Dynamic Optimization Problems – Updated Survey. In: GECCO workshop on Evolutionary Algorithms for Dynamic Optimization Problems; 2001, p. 27–30.

[2] Clarke DW, Mohtadi C, Tuffs PS. Generalized Predictive Control – Part I. The Basic Algorithm, Automatica; 1987 23(2):137–148.

[3] Filipič B, Juričić Đ. An Interactive Genetic Algorithm for Controller Parameter Optimization. In: R. F. Albrecht et al. (Eds.), Proc. of the International Conference on Artificial Neural Networks and Genetic Algorithms; 1993, p. 458–462.

[4] Fogarty TC, Vavak F, Cheng P. Use of the Genetic Algorithm for Load Balancing of Sugar Beet Presses. In: L. Eshelman (Ed.), Proc. of the Sixth International Conference on Genetic Algorithms; 1995, p. 617–624.

[5] Huang W, Lam HN. Using Genetic Algorithms to Optimize Controller Parameters for HVAC Systems, Energy Build; 1997, 26(3):277–282.

[6] Krink T, Ursem RK, Filipič B. Evolutionary Algorithms in Control Optimization: The Greenhouse Problem. In: L. Spector et al. (Eds.), Proc. of the third Genetic and Evolutionary Computation Conference (GECCO-2001); 2001, p. 440–447.

[7] Pohlheim H, Heißner A. Optimale Steuerung des Klimas im Gewächshaus mit Evolutionären Algorithmen: Grundlagen, Verfahren und Ergebnisse, Techincal report, Technische Universität Ilmenau; 1996.

[8] Ursem RK. Multinational GAs: Multimodal Optimization Techniques in Dynamic Environments. In: Proc. of the Second Genetic and Evolutionary Computation Conference (GECCO-2000); 2000, p. 19–26.

[9] Ursem RK, Krink T, Filipič B. A Numerical Simulator of a Crop-Producing Greenhouse, Techincal report no. 2002-01, EVALife, Dept. of Computer Science, University of Aarhus; 2002. www.evalife.dk

[10] Ursem RK, Krink T, Jensen MT, Michalewicz Z. Analysis and Modeling of Control Tasks in Dynamic Systems, IEEE Transactions on Evolutionary Computation (in press).