Optimal Control of Greenhouse Climate using a Short Time Climate Model and Evolutionary Algorithms

Hartmut Pohlheim

Technical University Ilmenau, Faculty of Informatics and Automation, Germany

Adolf Heißner

Institut für Gemüse- und Zierpflanzenbau Großbeeren/Erfurt, Germany

Overview

- Motivation
- Integrated Greenhouse climate model
- Optimization using Evolutionary Algorithms
- Experiments and Results
- Summary and Future work

Motivation

Understanding the processes in the greenhouse

By: development of complex models

- integrated model including
  - → greenhouse climate
  - → plant model (crop growth, yield)
  - → outside weather conditions
  - → control equipment (heating, ventilation, CO_2 injection)

- providing
  - → prediction of all inside climate conditions in a 15-60 minute time interval
  - → use for short time control tasks
    - ⇨ optimization of crop growth / yield
    - ⇨ prevention of stress

Motivation

Controlling the processes in the greenhouse

By: optimization of control

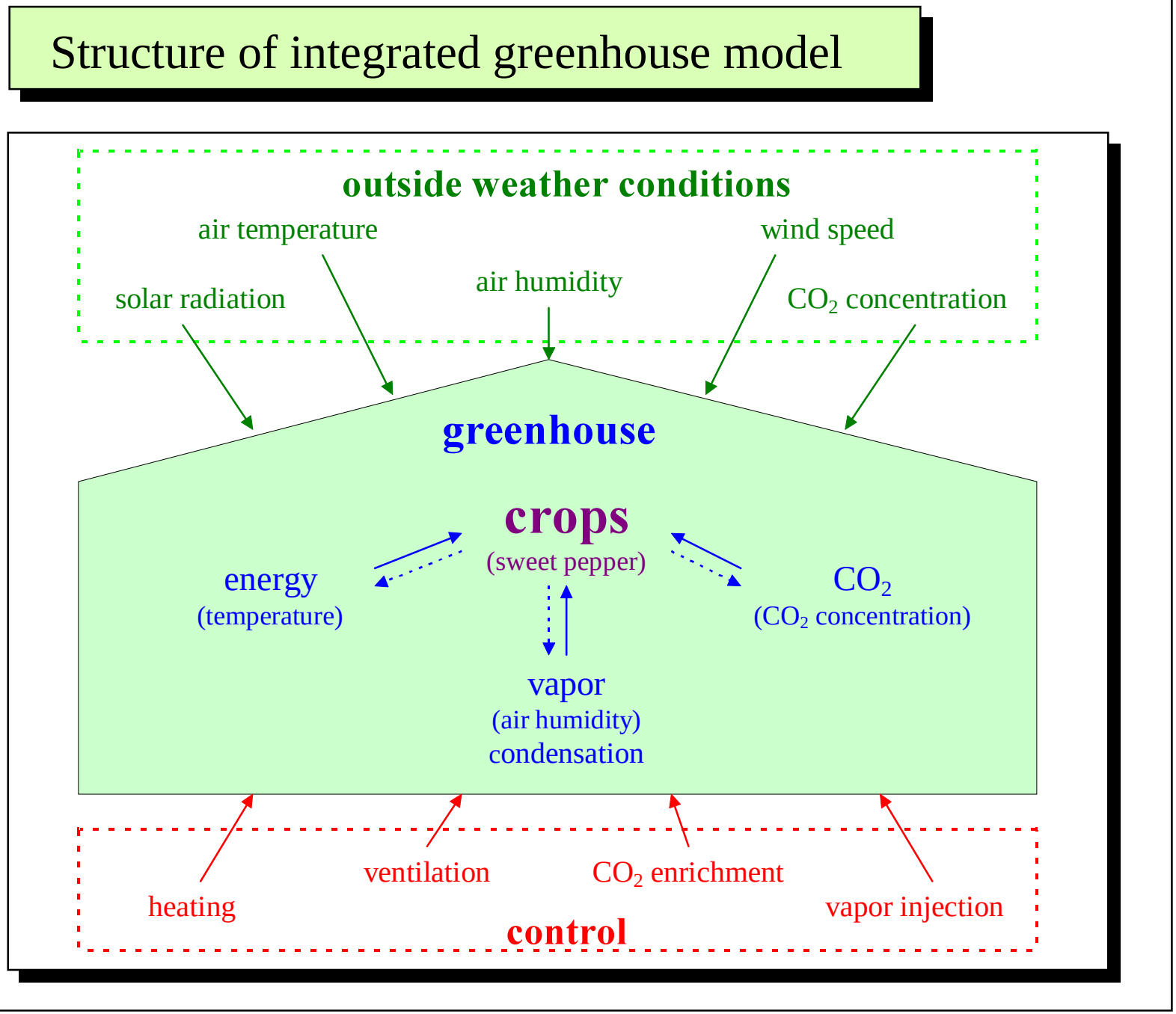
- goal: maximal yield under consideration of defined constraints

How:

- optimization of a dynamical system
- use of Evolutionary Algorithms
  - → direct and simple deployment
  - → incorporation of problem specific knowledge
  - → incorporation of constraints

Greenhouse model

Structure



TU ILMENAU

Informatics and Automation

Simulation of the integrated model

- 4 greenhouse balance equations
  - → energy balance
  - → vapor balance
  - → CO_2 balance
  - → condensation / evaporation
- 2 plant balance equations
  - → CO_2 exchange
  - → transpiration
- yield / biomass (derived from CO_2 exchange)
- profit (yield minus control costs)

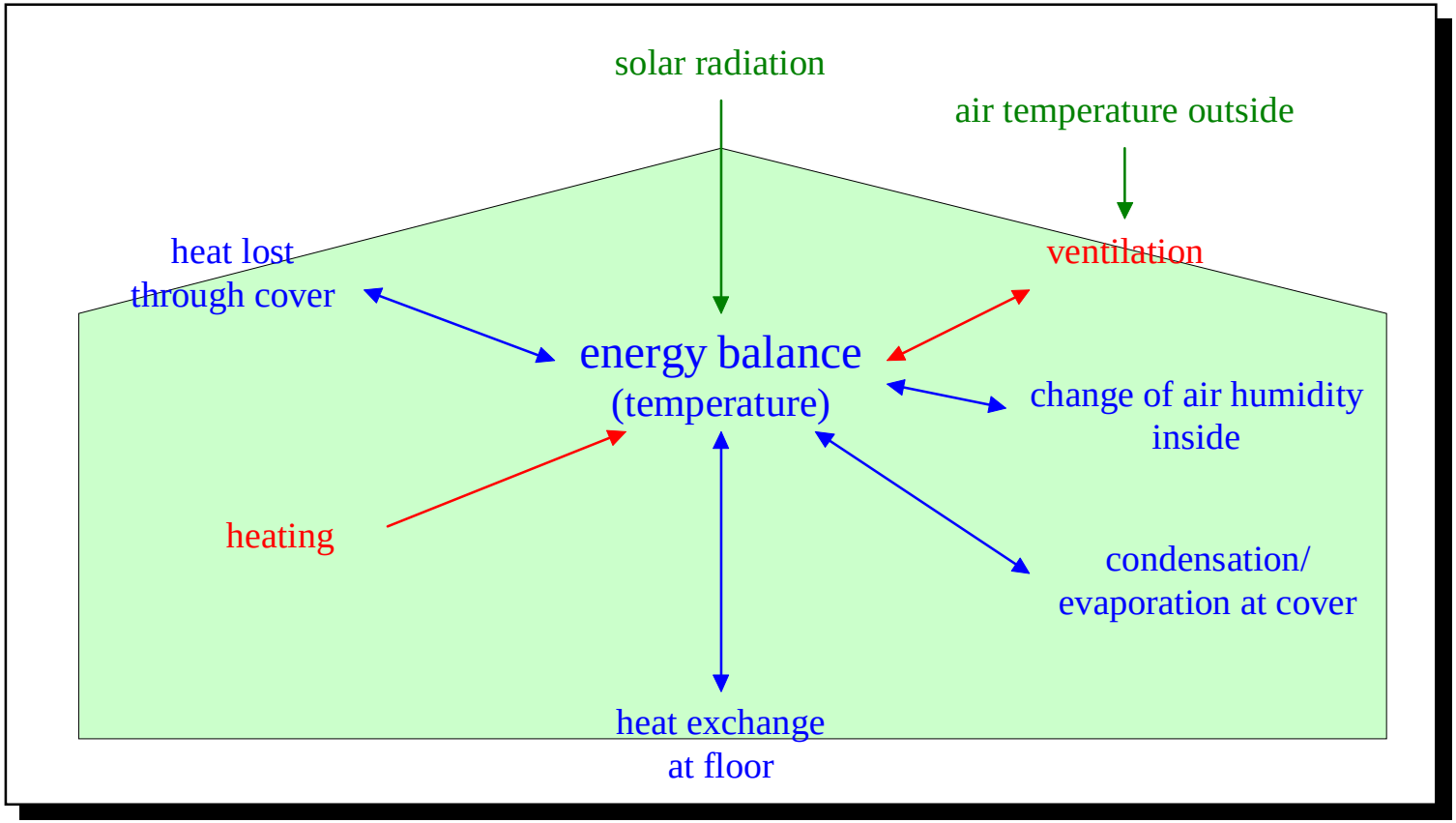
→ system of 8 first-order differential equations

- → simulated using Runge-Kutta methods of 4th(5th) order

Greenhouse model

Energy balance

Components of energy balance



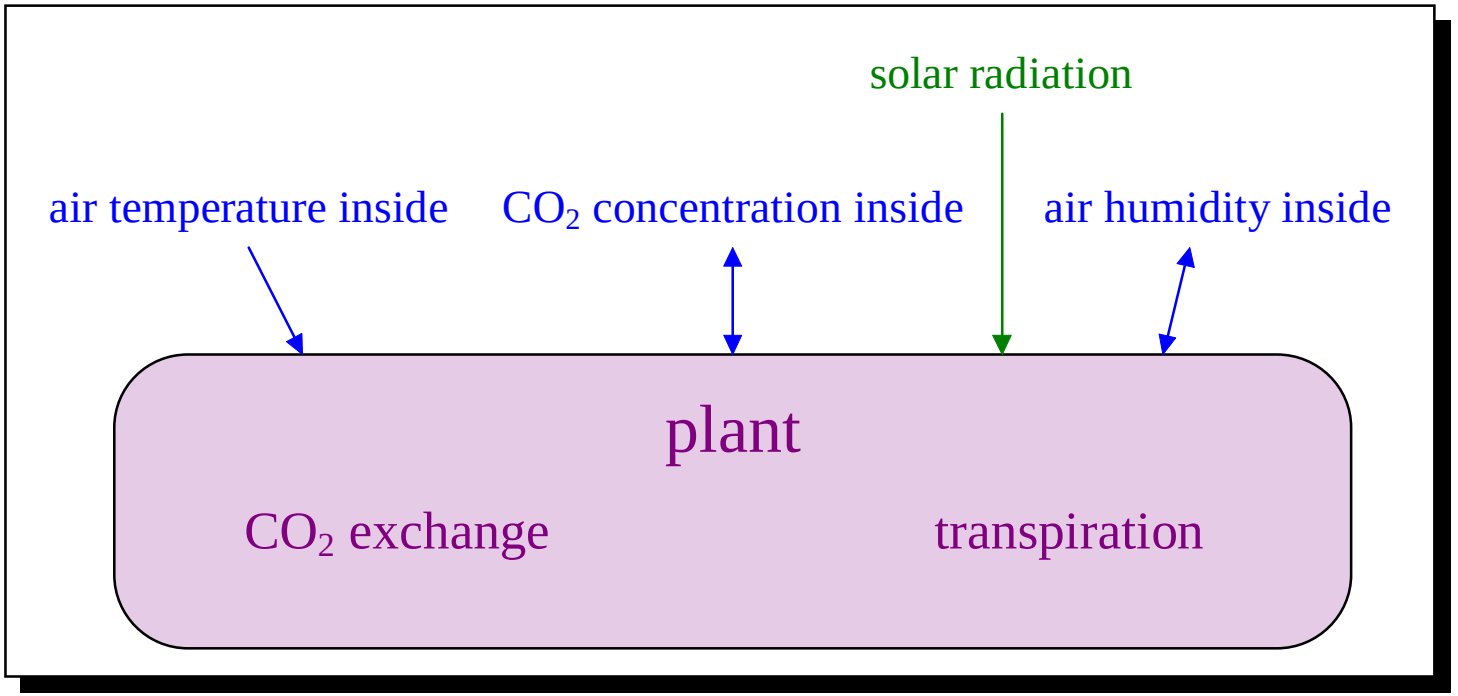
TU ILMENAU

Informatics and Automation

Greenhouse model

Plant model

Components of plant model



- CO_2 exchange and transpiration are computed relative to standard conditions
- use of parametric equations

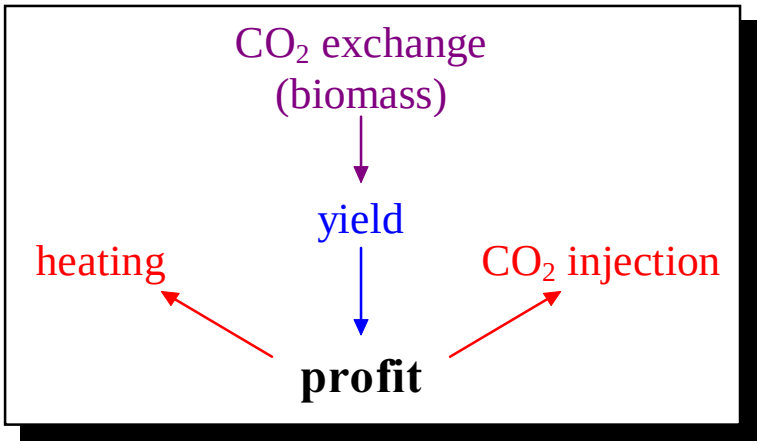
Motivation

- calculation of a control
  - → value of every control component over control time
  - → methods of optimization of dynamic systems
- What do we need?
  - → objective function (cost function)
  - → representation of solutions / individuals
  - → problem specific knowledge
- use of Evolutionary Algorithms
  - → straightforward application (compared to gradient methods)
  - → incorporation of problem specific knowledge
  - → incorporation of constraints

Objective function (cost function)

- maximization of profit

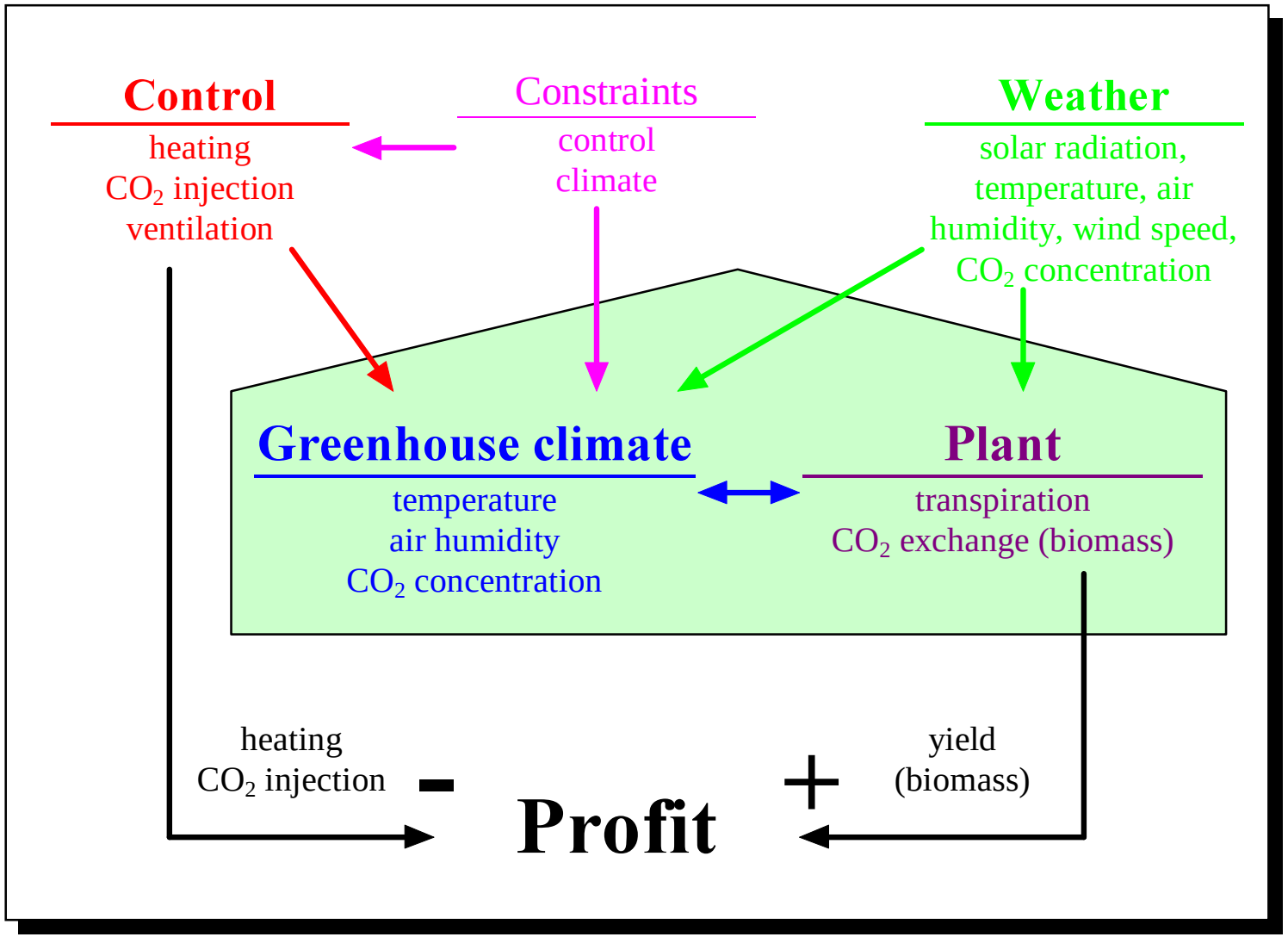
$$Cost = \int_{T_S}^{T_E} Profit\ dt + Penalty$$



- prevention of stress
  - → minimal/maximal temperature
  - → maximal CO_2 concentration
  - → minimal/maximal air humidity

$$Penalty = \sum_{i=1}^{NumConstr} \left(W_i \cdot \int_{T_S}^{T_E} \left| \left(Val_i - Constr_i \right) > 0 \right| dt \right)$$

Dependencies of profit calculation



Incorporation of domain specific knowledge

- initialization with known good control strategies
  - → good strategies are the result of optimization under standard (average) weather conditions
  - → good control values defined for every month and for each hour
  - → use multiple similar control strategies (previous / following months)
- changing domain of control variables
  - → restrict domain of control variables depending on time of year
    - ⇨ during winter heating is high, ventilation is low,
    - ⇨ during summer ventilation is higher than during winter, ...
  - → and time of day
    - ⇨ during night heating is higher than during day,
    - ⇨ during night ventilation is lower than during day, ...

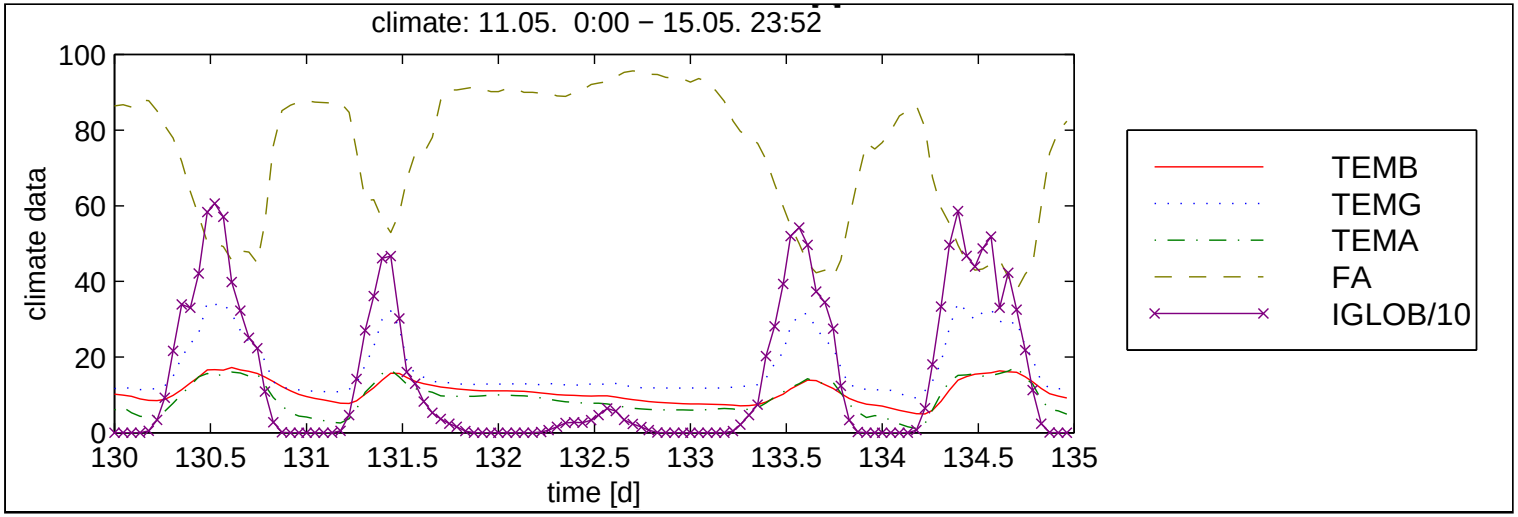
Experiments

Real data

Results under real world weather conditions

- optimization for 5 consecutive days in May 1995
- weather data of Großbeeren (Berlin), 1995

Weather data used

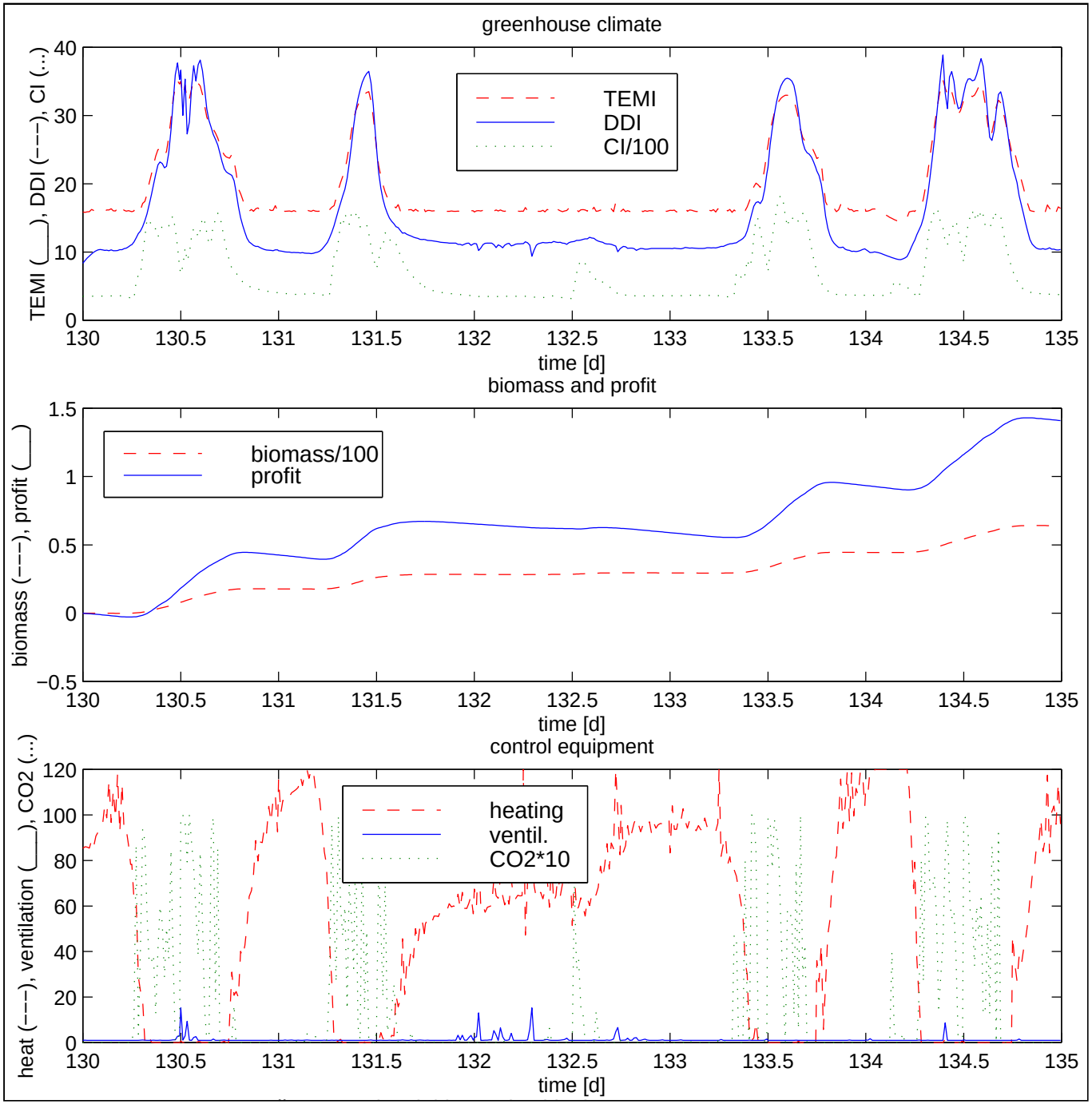


TEMB: floor temperature TEMG: cover temperature TEMA: air temperature outside

FA: air humidity outside IGLOB: solar radiation

Experiments

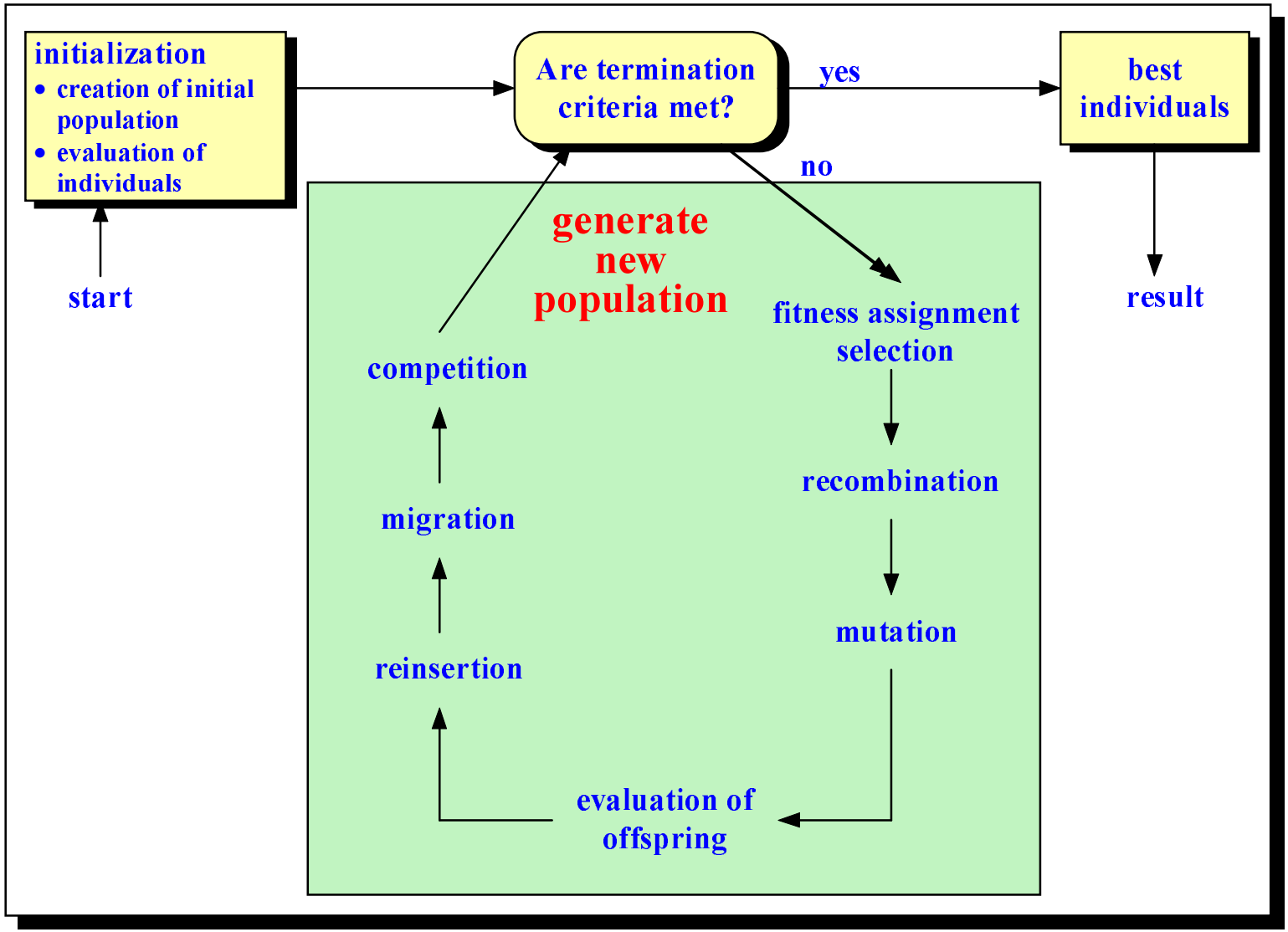
Real data



TU ILMENAU

Informatics and Automation

Structure of the evolutionary algorithm



Tools and algorithms used

- Genetic and Evolutionary Algorithm Toolbox for MATLAB
  - → Multi Strategy Competition EA
    - ⇨ 4 subpopulations with 50 individuals each
    - ⇨ linear ranking, truncation selection, generation gap 0.9 (elitest),
    - ⇨ discrete and line recombination,
    - ⇨ real valued mutation (different range and precision),
    - ⇨ unrestricted migration, competition between subpopulations
    - → globally oriented search strategy (rough and fine search combined)
  - → Evolutionary Strategy EA
    - ⇨ 3 subpopulation with 2 individuals each
    - ⇨ every individual produces 6 offspring (generation gap: 6)
    - ⇨ best two offspring replace parents
    - ⇨ no recombination
    - ⇨ mutation by mutation operator of an evolution strategy
    - → local search strategy (adaptation of mutation step sizes)

Summary

Advantages

- presentation of an integrated greenhouse model incl.
  - → greenhouse climate
  - → plant growth
  - → control equipment
  - → outside weather conditions
- use for short time-scale prediction tasks
- optimization using Evolutionary Algorithms
- employment of different objectives
  - → profit maximization, prevention of stress, constraints
- ready for online control
- incorporation of problem specific knowledge

Summary

Future work

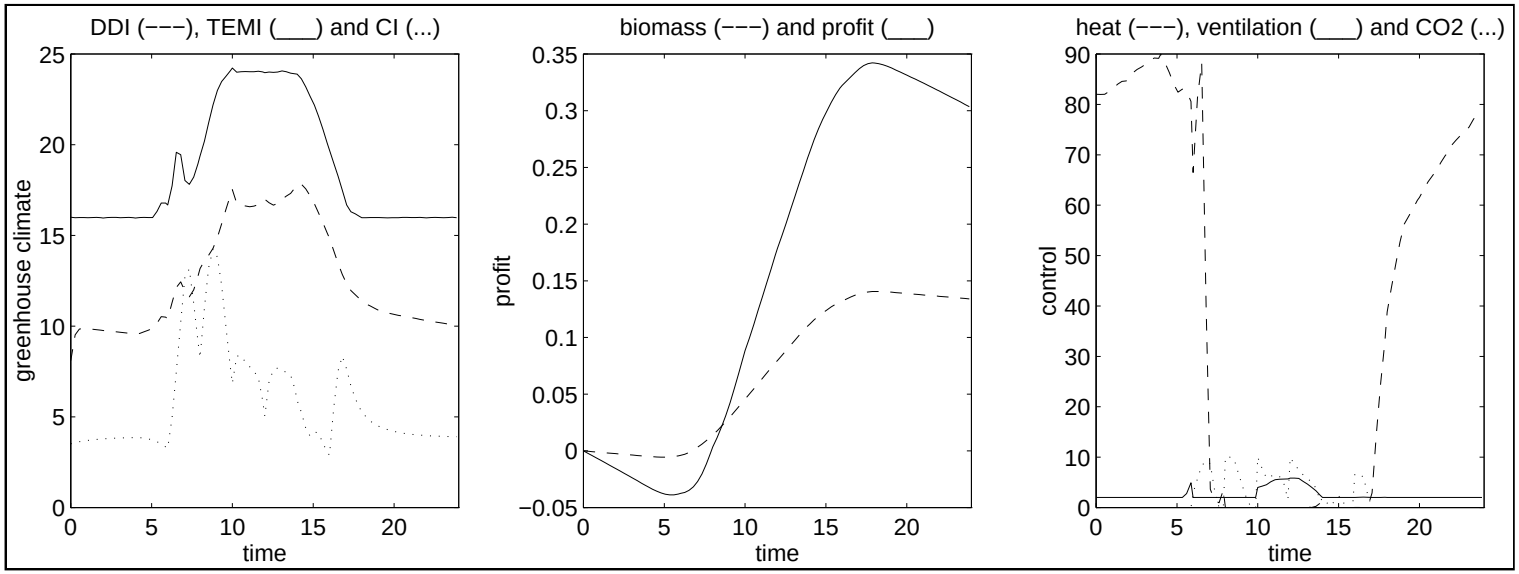
- new and enhanced plant models
  - → tomato, cucumber, sweet pepper, radish
- comparison with different control strategies
  - → constant set points for greenhouse climate
- coupling with long timescale control strategies
  - → long time-scale: high level strategy,
  - → short time-scale: low level strategy
- definition of additional objectives for optimization
- multiobjective ranking of different objectives
- enhancements to the evolutionary algorithm
  - → refinement of multiple strategies for subpopulations
  - → use of competing subpopulations
  - → employment of problem specific genetic operators

Experiments

Typical days

Optimal control for typical days

- optimization for April
- heating high during night (lower temperature constraint)
- ventilation around noon (upper temperature constraint 24°C)
- CO_2 injection on during day (with no or little ventilation)



TU ILMENAU

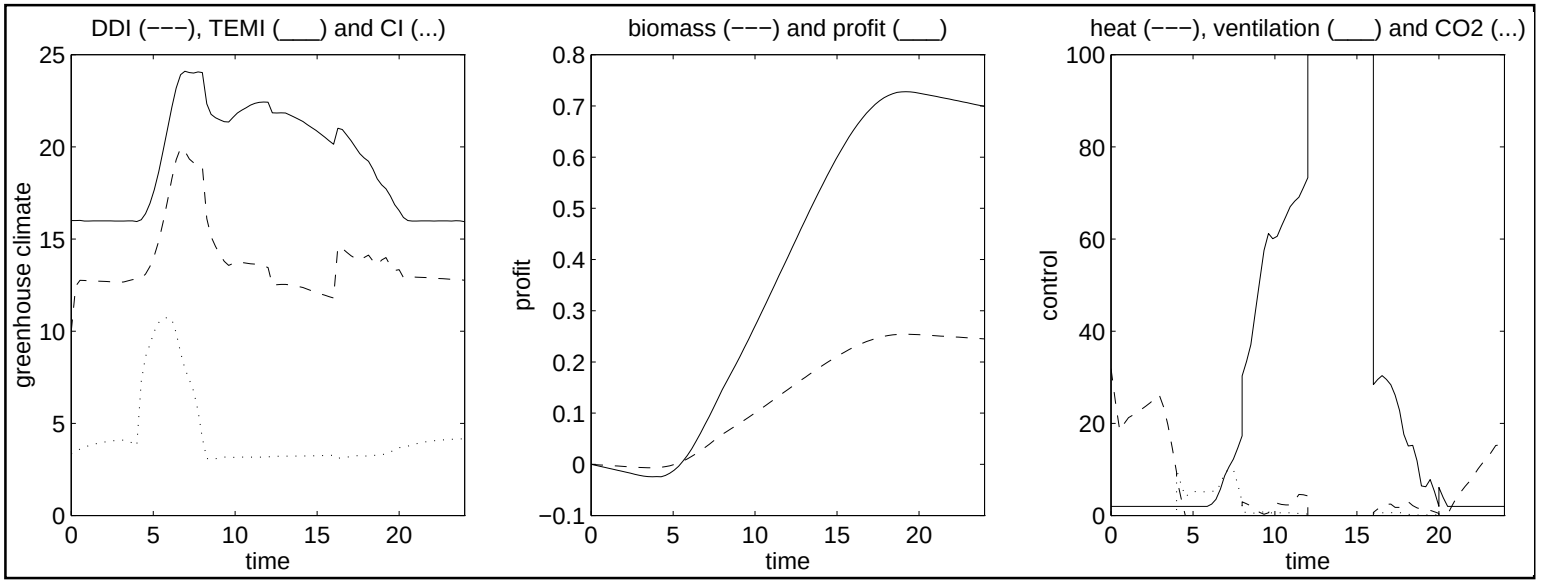
Informatics and Automation

Experiments

Typical days

Optimal control for typical days

- optimization for June
- heating on during night (lower temperature constraint)
- ventilation on during day
  (upper temperature constraint, 20°C better than 24°C)
- CO_2 injection on during morning (before ventilation opens)



TU ILMENAU

Informatics and Automation

Experiments

Changing prices

Optimal control under changing prices

- prices of heating (energy) and CO_2 were changed
  - → 3 times as high as standard price
  - → $^1/_3$ of standard price
- optimization for February
- high prices
  - → biomass: 1.1 g/(m^2·h) profit: -0.09 Pf/(m^2·h)
  - → heating as low as possible (lower temperature constraint)
  - → nearly no CO_2 injection
- low prices
  - → biomass: 1.5 g/(m^2·h) profit: 0.03 Pf/(m^2·h)
  - → heating higher than keeping temperature above 16°C
  - → during daytime CO_2 injection

TU ILMENAU

Informatics and Automation

Experiments

Comparisons

Comparison

previously published solutions

- use of models and simulation to find better set points
  - → day/night temperature
  - → night CO_2 enrichment
- no immediate reaction to changing conditions (prices)

new integrated model and evolutionary algorithms

- optimal control of greenhouse climate
- applicable for different objectives
- immediate response to changing conditions
- use for online control

TU ILMENAU

Informatics and Automation

Representation of individuals

- individuals represent control of simulation period
  - → use of 3 controls (heating, ventilation, CO_2 enrichment)
  - → controls discretized at equidistant points (first order hold)
  - → control step every 15 minutes
  - → simulation period: 2-12 hours

$$NumVar = \left(\frac{SimTime}{ControlStep} + 1 \right) \cdot NumControl; \qquad \begin{array}{l} ControlStep = 0.25h \\ NumControl = 3 \end{array}$$

- divide longer optimization periods into smaller pieces
  - → end states of one period are start states of next period
  - → arbitrary long time periods can be calculated
  - → size of optimization problem is adjustable to available computing power