EVOLUTIONARY COMPUTATION AND MULTIOBJECTIVE OPTIMIZATION

Gary G. Yen, *FIEEE*, *FIET, FIAPR*

gyen@okstate.edu

***Regents Professor*, Oklahoma State University**

四川大学高端外籍讲座教授

Summer Course at

四川大学计算机学院

Day One of EIGHT, June 27, 2022

Instructor

Professor Gary G. Yen, Ph.D., *FIEEE*, *FIET, FIAPR*

Phone: +1-405-744-7743; Email: *gyen@okstate.edu*

Gary G. Yen received his Ph.D. degree in electrical and computer engineering from the University of Notre Dame (圣母大学) in 1992. He is currently a Regents Professor in the School of Electrical and Computer Engineering, Oklahoma State University (俄克拉荷马州立大学). Before joined OSU in 1997, he was with the U.S. Air Force Research Laboratory in Albuquerque. His research is supported by the DoD, DoE, EPA, NASA, NSF, and Process Industry. His research interest includes control optimization, computational intelligence, conditional health monitoring, artificial intelligence/machine learning and their industrial/defense applications. He is an IEEE Fellow, IET Fellow and IAPR Fellow.

Dr. Yen was an associate editor of the *IEEE Control Systems Magazine*, *Automatica*, *IEEE Transactions on Control Systems Technology*, *IEEE Transactions on Systems, Man and Cybernetics, Part A and B* and *IEEE Transactions on Neural Networks*. He is currently serving as an associate editor for the *IEEE Transactions on Evolutionary Computation* (IF 11.554), *IEEE Transactions on Cybernetics*, *IEEE Transactions on Emerging Topics on Computational Intelligence* and *IEEE Transactions on Artificial Intelligence*. He served as the General Chair for the *2006 and 2016 IEEE World Congress on Computational Intelligence* held in Vancouver, Canada. Dr. Yen served as Vice President for the Technical Activities in 2005-2006 and ***President*** in 2010-2011 of the IEEE Computational intelligence Society and is the ***Founding Editor-in-Chief*** of the *IEEE Computational Intelligence Magazine* 2006-2009 (IF 11.356).

In 2011, he received the Andrew P Sage Best Transactions Paper award from IEEE Systems, Man and Cybernetics Society. In 2013, he received Meritorious Service award from IEEE Computational Intelligence Society.

Teaching Assistant

Shanlin Jiang (江姗霖)
LeoDJiangshanlin@outlook.com

Master student of Dr. Zhenan He

Housekeeping

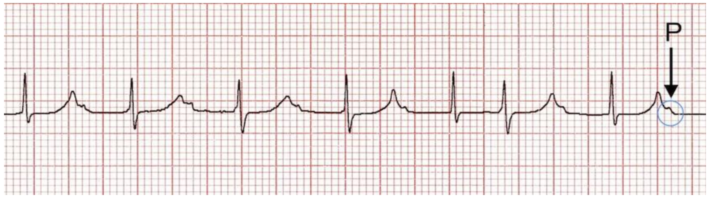
- Your final grade will be determined by your ***timely*** turn-in homework and your ***participation*** to the lectures.
- "Asking questions" or "sharing comments" during the lectures will be counted toward your participation.
- If you have any aspiration to engage in research in this subject matter, please contact me directly.

- Lecture notes in PDF will be made available to you before each lecture. You are highly encouraged to bring with you a notebook, iPad, tablet PC, or smart phone that would allow you to view the lecture slides. In the worst-case scenario, you can have lecture notes printed out before you come to each lecture.

- Plan for Lecture on Wednesday, June 29
  - Push down to Friday, July 1st
    - Will the same classroom be available?
  - Or I will ask Prof. Zhenan He to teach something on my behalf

Case Study 1:
Early Time-Series Classification

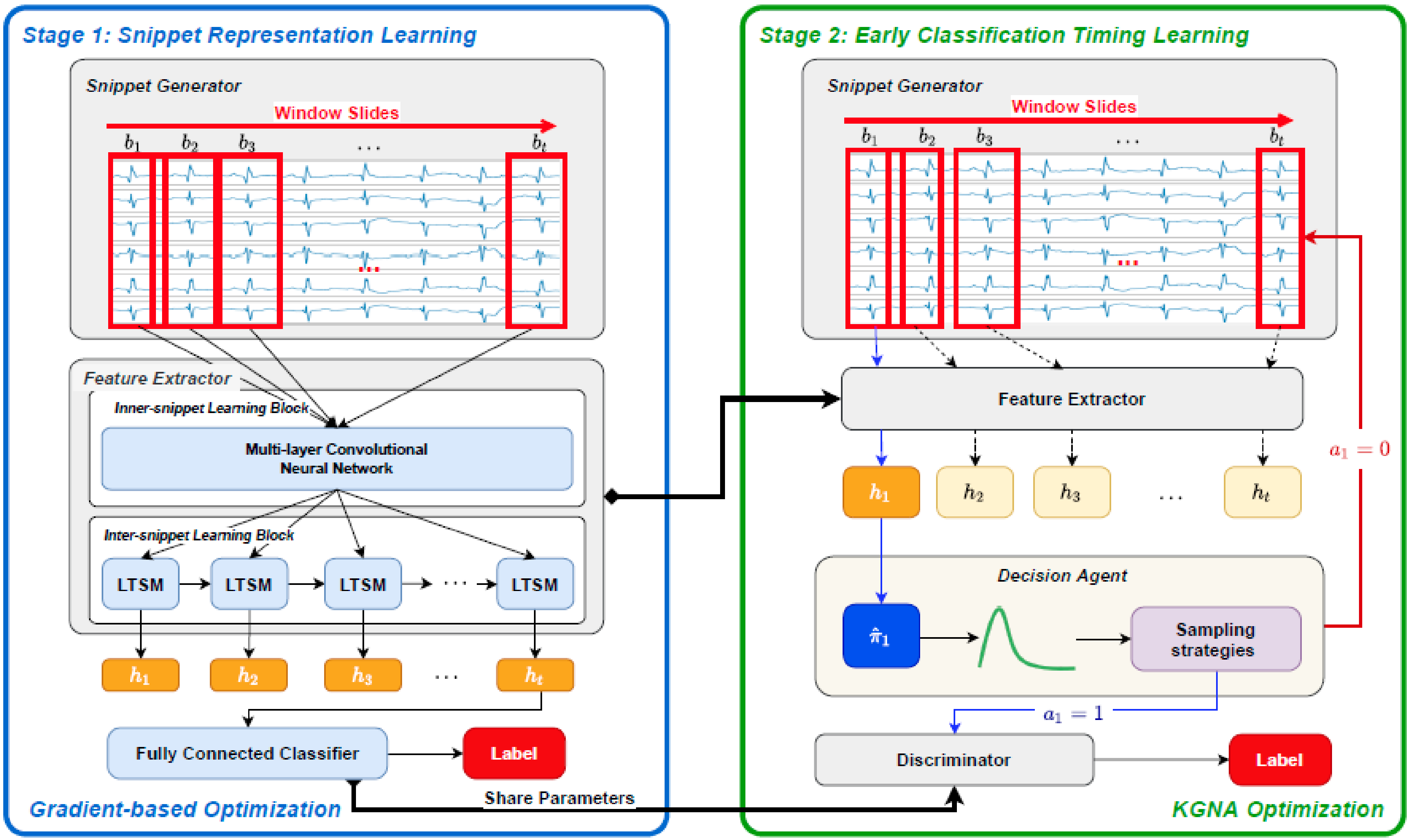
- Early time series classification predicts the class label of a given time series before it is completely observed.



- **Motivation:** In time-critical applications, such as arrhythmia monitoring in ICU/ER, early treatment contributes to the patient's fast recovery, and early warning could even save lives. Hence, in these cases, it is worthy of trading some extent classification accuracy in favor of earlier decisions when the time series data is collected over time.
- **Approach:** We propose a novel deep reinforcement learning-based framework, Snippet Policy Network V2 (SPN-V2), for long and varied-length multi-lead ECG early classification. The proposed SNP-V2 contains two main components, which are snippet representation learning and early classification timing learning.

- The snippet representation learning is proposed to encode *inner-snippet spatial correlations* and *inter-snippet temporal correlations* into the hidden representations of the sub-segment (snippet) of the input ECG. Early classification timing learning aims to learn a decision agent to classify the time series early and accurately.
- To optimize the proposed framework, we design a novel *knee-guided neuroevolution algorithm* to solve cardiovascular diseases' early classification problem, automatically optimizing the proposed SPN-V2 regarding the trade-off between accuracy and earliness.
- Through a series of experiments on two public ECG classification datasets, the results demonstrate that the superiority of our proposed model compared with state-of-the-art methods, which gains up to 7% in terms of precision, recall, accuracy, F1-score, and harmonic mean.

"Snippet Policy Network V2: knee-guide neuroevolution for multi-lead ECG early classification," Huang Y., Yen G.G., and Tseng V.S., ***IEEE Transactions on Neural Networks and Learning Systems***, 2022, in early access.



Stage 1: Snippet Representation Learning
Snippet Generator
Window Slides
b_1 b_2 b_3 ... b_t
Feature Extractor
Inner-snippet Learning Block
Multi-layer Convolutional Neural Network
Inter-snippet Learning Block
LTSM LTSM LTSM ... LTSM
h_1 h_2 h_3 ... h_t
Fully Connected Classifier
Label
Gradient-based Optimization
Share Parameters
Stage 2: Early Classification Timing Learning
Snippet Generator
Window Slides
b_1 b_2 b_3 ... b_t
Feature Extractor
h_1 h_2 h_3 ... h_t
$a_1 = 0$
Decision Agent
$\hat{\pi}_1$
Sampling strategies
$a_1 = 1$
Discriminator
Label
KGNA Optimization

1. COMPUTATIONAL INTELLIGENCE

计算智能

Wikipedia

Computational intelligence **(CI)** is a set of ***nature-inspired computational methodologies*** and approaches to address complex real-world problems to which traditional approaches, i.e., first principles modeling or explicit statistical modeling, are ineffective or infeasible. Many such real-life problems are ***not*** considered to be ***well-posed problems mathematically***, but nature provides many counterexamples of biological systems exhibiting the required function, practically. … Traditional models also often fail to handle ***uncertainty, noise and the presence of an ever-changing context***. Computational Intelligence provides solutions for such and other complicated problems and inverse problems. It primarily includes artificial neural networks, evolutionary computation and fuzzy logic. An important aspect of Computational Intelligence is adaptivity which is covered by the fields of machine learning and computational neuroscience. In addition, CI also embraces biologically inspired algorithms such as swarm intelligence and artificial immune systems.

Definition of CI

Any biologically, naturally, and linguistically motivated computational paradigms include, but not limited to,

neural network, *connectionist machine*, *fuzzy system*, *evolutionary computation*, *autonomous mental development*,

and *hybrid intelligent systems* in which these paradigms are contained.

Any nature-inspired computational paradigms for problem solving…

Coined by the *IEEE Computational Intelligence Society* (*1990, 2001, 2005*) Credited to Jim Bezdek, USA

Evolutionary Computation

The Origin

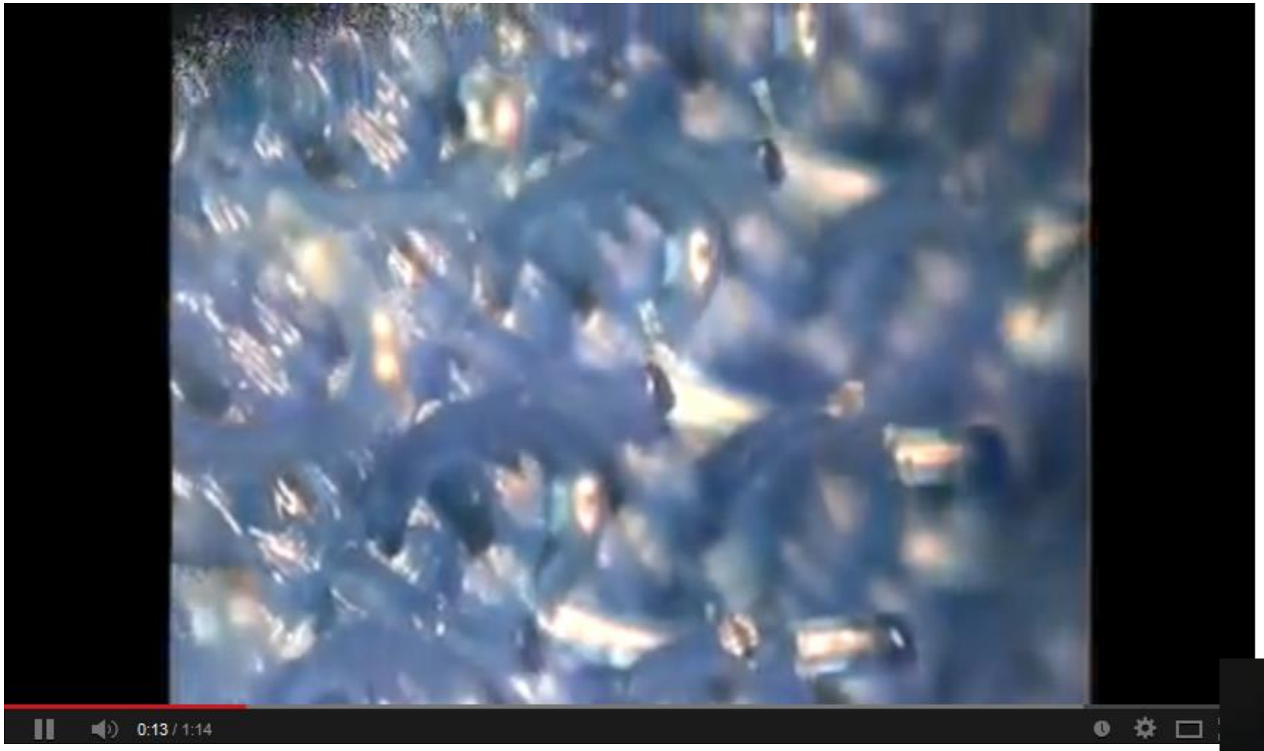
Many pioneering scientists, including Newton and Maxwell, were motivated by a quest to discover the art and order in creation- to know the mind of God through study of His creations.

Nearly all inventions have a counterpart in or are an extension of nature.

- thermonuclear explosions occur in the stars
- pulse modulation occurs in the human nervous system
- bats have sonar and dolphin pings serve as a subterranean telephone

- Nature/biology inspires invention
- Engineering uses science and mathematics to emulate and extend nature
- As the bird motivates air flight, so does human/biological intelligence motivates study of advanced computational paradigms.

A Practical Example- Velcro



Burs

魔术贴

Common Characteristics

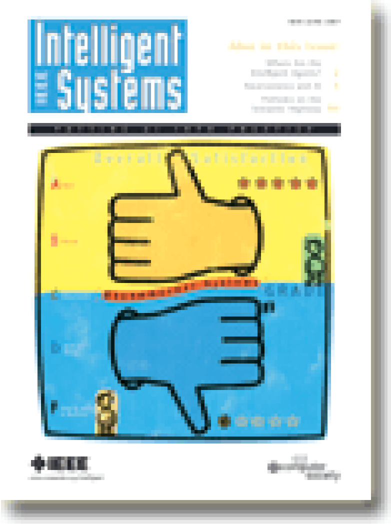
- Biologically motivated behavior such as learning, reasoning, or evolution (*in the sense of approximation*)
- Parallel, distributed information processing
- Mysterious power under real-world complications

- Lack of qualitative analysis
- Non-repeatable outcomes
- Stochastic nature

Is this a panacea for our modern-day problems?
灵丹妙药

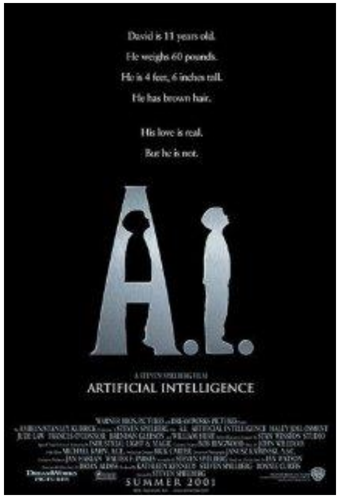
CI vs. Intelligent Systems (IS)

- IS covers on all aspects of artificial intelligence, focusing on the development of the latest research into practical, fielded *applications*.
- CI, on the other hand, is a collective effort in emerging, *fundamental* computational paradigms.



CI vs. Old School AI

- CI depends upon numerical data supplied by manufacturers and does not rely on “knowledge.”
- AI, on the other hand, uses “knowledge tidbits” and these knowledge is derived from human expert.
- The knowledge or intelligence exhibited from CI is *self-emerging and spontaneous* as opposed to *man-made and artificial* from AI.



Paradigm Shift…

Theory

Existence/Uniqueness of a Solution;
Necessary/Sufficient conditions for Local Extreme

Experimentation

Scientific discovery from Renaissance through the age of Enlightment

Computer Simulation

Model-based optimization based on Statistics/Computing power

Data Analytics

Shift from providing simple analysis tools to providing full-fledged knowledge discovery frameworks

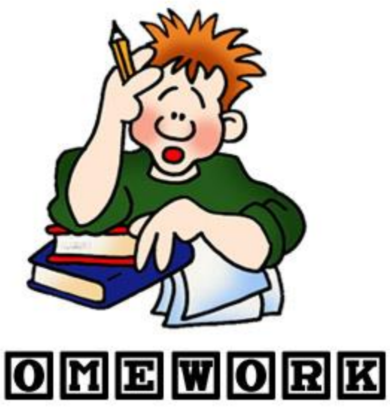
Machine learning to distill data science models that can automatically extract relationships

Theory-Guided Data Science

Homework #1

What could be a potential biological behavior/metaphor that could be transformed into a computational intelligence paradigm?

GA, ES, EP, DE and PSO



- Due: next Lecture on 6/28/22, before noon
- Please email your answer to TA


- As an example, the ***bat algorithm***, inspired by the echolocation behavior of microbats with varying pulse rates of emission and loudness, is metaheuristic algorithm for global optimization.

Yang, X. S. (2010). "A New Metaheuristic Bat-Inspired Algorithm," *Studies in Computational Intelligence*. **284**: 65–74.

- As another example, inspired by the leadership hierarchy and hunting mechanism, ***grey wolf algorithm*** embodies four types of grey wolves (alpha, beta, delta, and omega) for division of labor to explore the search space through searching for prey, encircling prey, and attacking prey.

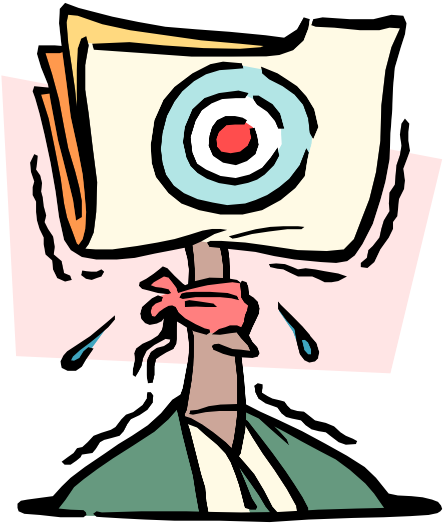
Mirjalili, S, Mirjalili, S.M. and Lewis, A. (2014). “Grey Wolf Optimizer". *Advances in Engineering Software*. **69**: 46-61.

Critical Message Conveyed

Any *nature-inspired* computational paradigms for problem solving…
is

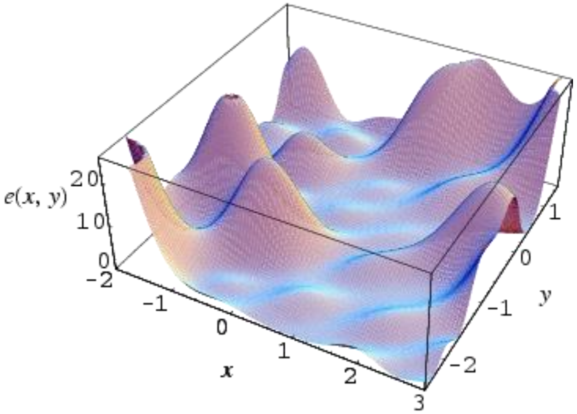
Computational Intelligence

Q&A



2. OPTIMIZATION PROBLEMS

优化问题



Optimization Problem Formulation

3 basic ingredients…

- a set of objective functions,
- a set of decision variables,
- a set of equality/inequality constraints.

The problem is

to search for the values of the decision variables that minimize or maximize the objective functions while satisfying the constraints…

An example…

How to partition 1,000 circuits into five regions on a microchip and confirm to layout constraints?

5^1000 = 9.33263619E698

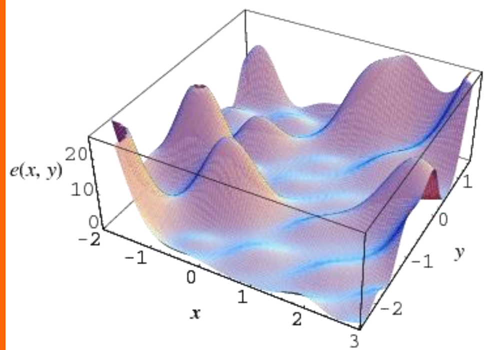
Mathematical Definition

- Mathematical model to formulate the optimization problem

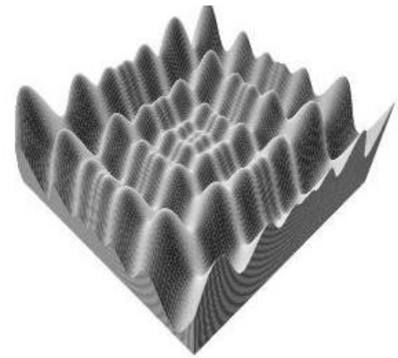
Objective vectors | **Decision vectors** | **Environment States** | **Equality constraints** | **Inequality constraints** | **Variable bounds**

$$\min_{\mathbf{x}\in\Re^{\mathbf{n}}}\{\mathbf{y} \cong \mathbf{f}(\mathbf{x}, e) : \mathbf{h}(\mathbf{x}, e) = 0, \mathbf{g}(\mathbf{x}, e) \le 0, x^L \le x \le x^U\}$$

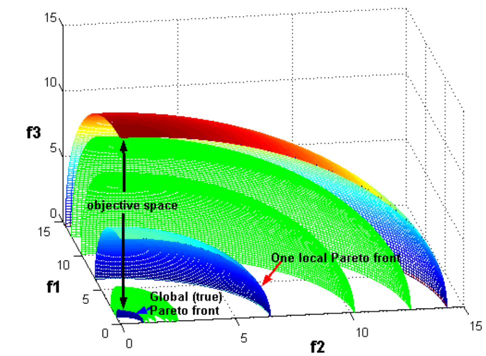
- o Design Variables: decision and objective vector
- o Constraints: equality and inequality
- o Greater-than-equal-to inequality constraint can be converted to less-than-equal-to constraint by multiplying -1
- o Objective Function: maximization can be converted to minimization due to the *duality principle* $\max\ f(x) = \min\ (-f(x))$



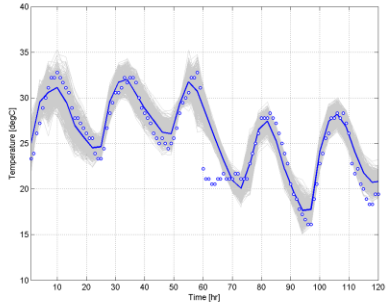
Single-objective Optimization



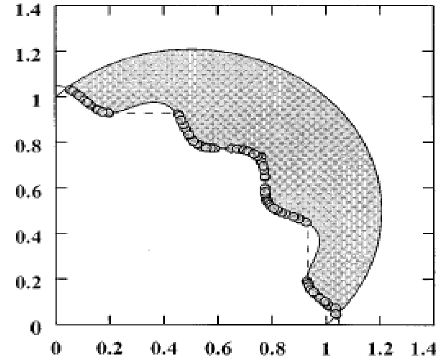
Multimodal Optimization



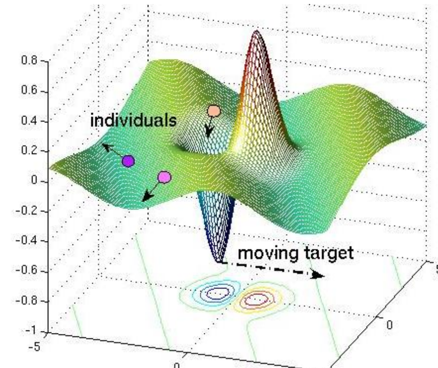
Multi-objective Optimization



Uncertainty Optimization



Constrained Optimization

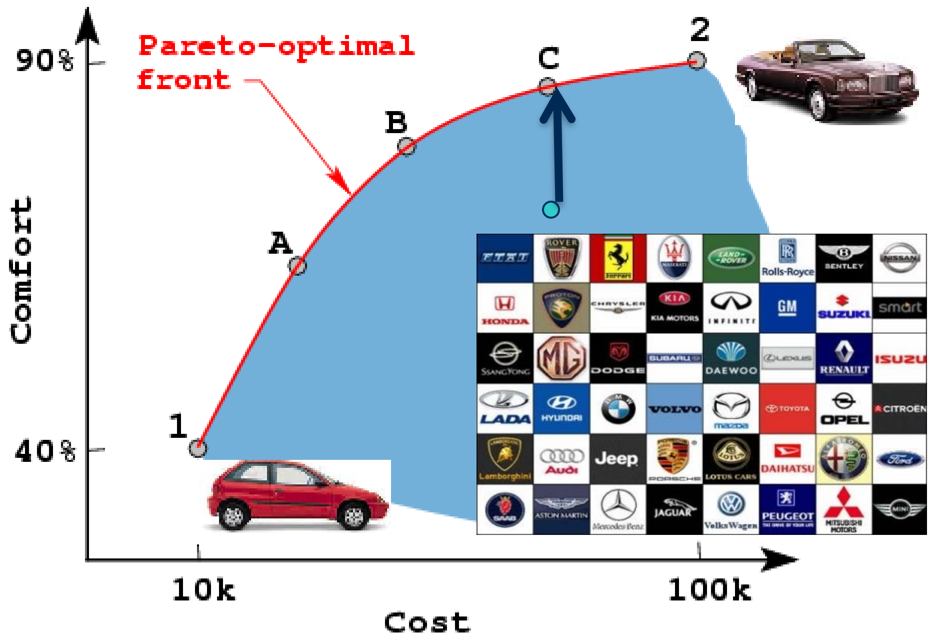


Dynamic Optimization

Multiobjective Optimization

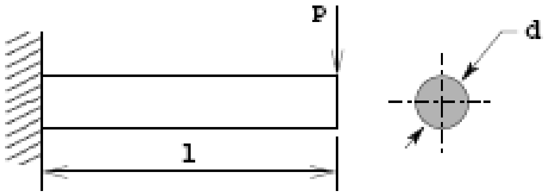
- Optimization problems involve more than one objective functions
- Very common, yet difficult problems in the field of science, engineering, and business management
- Nonconflicting objectives: achieve a single optimal solution satisfies all objectives simultaneously → SOPs
- ***Competing objectives***: cannot be optimized simultaneously
- MOP– find ***a set*** of "acceptable"– maybe only suboptimal for one objective– solutions is our goal
- In operation research/management terms - *multiple criterion decision making* (MCDM)

Buying an Automobile



- Objective = reduce cost, while maximize comfort
- Which solution (1, A, B, C, 2) is best ???
- No solution from this set makes both objectives look better than any other solution from the set
- No single optimal solution
- Trade off between conflicting objectives- cost and comfort

Engineering Design



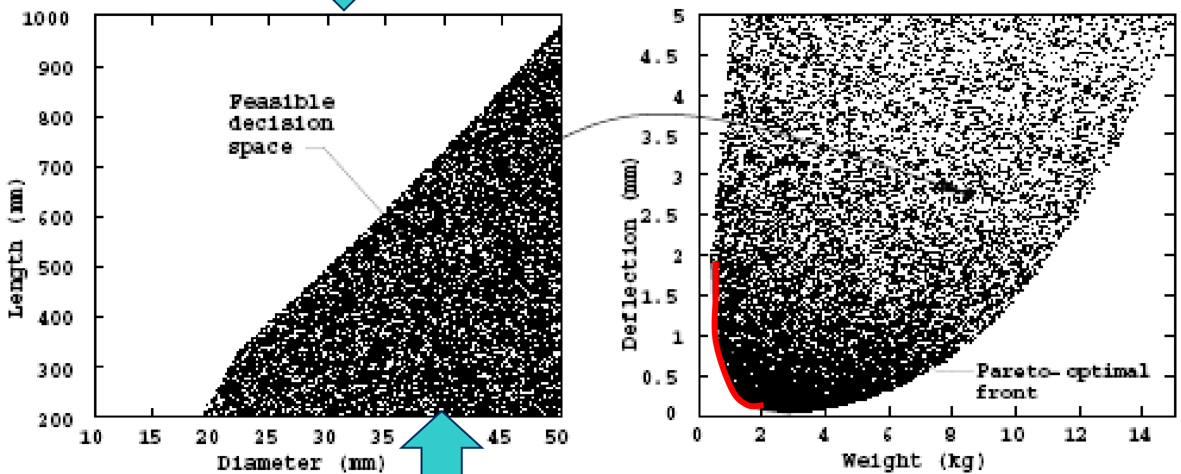
$$\text{Minimize} \quad f_1(d,l) = \rho \frac{\pi d^2}{4} l$$

$$\text{Minimize} \quad f_2(d,l) = \delta = \frac{64Pl^3}{3E\pi d^4}$$

$$\text{subject to} \quad \sigma_{\max} \le S_y$$

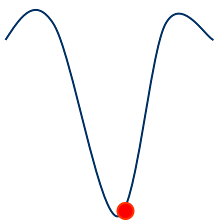
$$\delta \le \delta_{\max}$$

Infesaible region



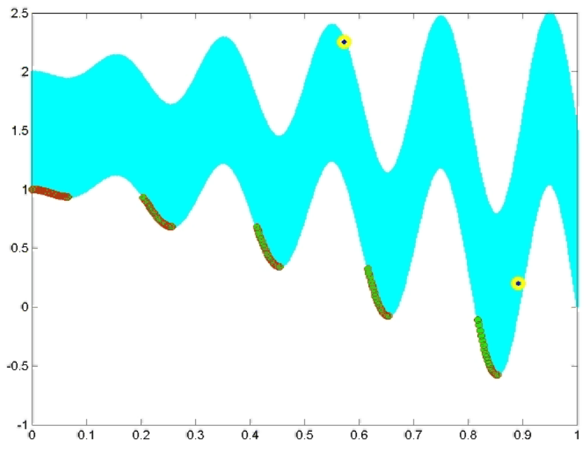
Feasible region

- Single objective optimization is a degenerate case of MOP

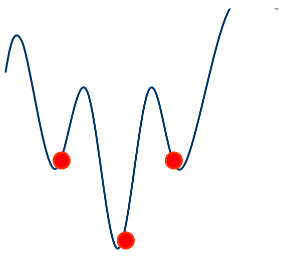


↔

MOP is not a simple extension of SOP



- Multi-modal optimization is a special case of MOP



Optimization Approaches

- applicable only to SOP
- search in a large space
- Derivative-based optimization (gradient based)
  - Capable of determining “search directions” according to an objective function’s derivative information
  - e.g., steepest descent method; Newton’s method; Newton-Raphson method; and those in between such as conjugate gradient method or Levenberg-Marquardt method
- Derivative-free optimization
  - e.g., simulated annealing; genetic algorithm; random search method; downhill simplex search
  - For both continuous and discrete optimization problems
  - For combinatorial optimization problems

Common Characteristics

- *Derivative freeness*
  - these methods do not need functional derivative information to search for a set of parameters that minimize a given objective function. Instead, they rely exclusively on repeated evaluations of the objective function. The subsequent search direction after each evaluation follows certain heuristic guidelines.

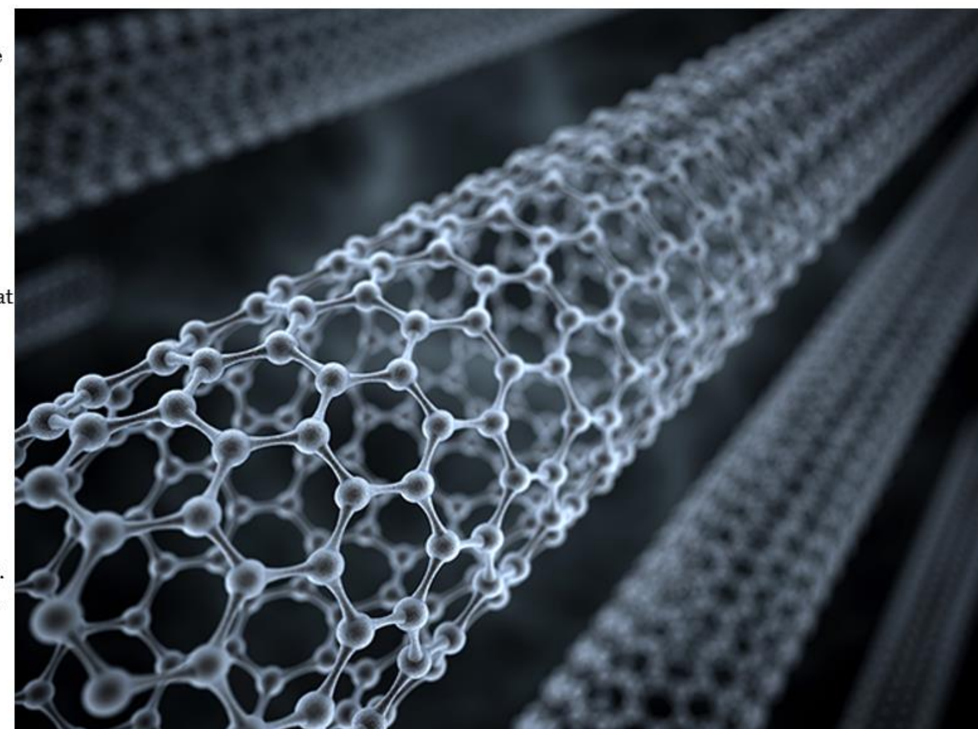
- *Intuitive guidelines*
  - The guidelines followed by these search procedures are usually based on “simple,” “heuristic” concepts. Some of these concepts are motivated by so-called “nature’s wisdom,” such as evolution and thermodynamics.
- *Slowness*
  - Without using derivatives, these methods are bounded to be generally slower than derivative-based optimization methods for continuous optimization problems.

- ***Flexibility***
  - Derivative freeness also relieve the requirement for differentiable objective functions, so we can use as complex an objective function as a specific application might need, without sacrificing too much in extra coding and computation time.

- *Randomness*
  - All of these methods (except standard downhill simplex) are stochastic, which means that they all use random number generators in determining subsequent search directions. This element of randomness usually gives rise to the overly optimistic view that these methods are “global optimizers” that will find a global optimum given enough computation time. In theory, their random nature does make the probability of finding an optimal solution nonzero over a fix amount of computation time. In practice, however it might take a considerable amount of computation time, if not forever, to find the optimal solution of a given problem.

A True Random-Number Generator Built From Carbon Nanotubes

By Amy Nordrum (/author/nordrum-amy)
Posted 9 Aug 2017 | 19:00 GMT



Researchers have built a true random number generator that they say could improve the security of printed and flexible electronics. They made it from a static random-access memory cell printed with a special ink containing carbon nanotubes (http://spectrum.ieee.org/semiconductors/devices/how-well-put-a-carbon-nanotube-computer-in-your-hand). The memory cell uses fluctuations in thermal noise to generate random bits.

Generating random numbers within an electronic device is critically important because random numbers are the basis for encryption keys that keep personal devices secure. Many electronics contain hardware components designed for this exact purpose.

It's also possible to generate random numbers through software. But software-based random number generators (https://www.khanacademy.org/computing/computer-science/cryptography/crypt/v/random-vs-pseudorandom-number-generators) are considered "pseudorandom." They start with an original number, or seed, and apply a mathematical equation to generate a string. The resulting pattern is not entirely random, and hackers can replicate it if they figure out the seed.

Hardware-based "true" random number generators are therefore considered the gold standard for security, but they can be bulky, rigid,

- ***Iterative nature***
  - Unlike the linear least-squares estimator, these techniques are iterative in nature, and we need certain stopping criteria to determine when to terminate the optimization process
  - Let k denote an iterative count and

    f_k denote the best objective function obtained at count k.
  - Stopping criteria:
    - computation time: a designated amount of computation time or number of iterations/function evaluation
    - optimization goal: $f_k < \delta$
    - Minimization improvement: $f_k - f_{k-1} < \delta$
    - Minimal relative improvement: $\frac{f_k - f_{k-1}}{f_{k-1}} < \delta$

- ***Analytic opacity***
  - It is difficult to do analytic studies of these methods, in part because of heir randomness and problem–specific nature. Therefore, most of our knowledge about them is based on empirical studies.

Traditional Approaches

- Bundle methods
- Conjugate gradient method
- Ellipsoid method
- Frank–Wolfe method
- Gradient descent aka steepest descent or steepest ascent
- Interior point methods
- Line search
- Nelder-Mead method aka the Amoeba method
- Newton's method
- Quasi-Newton methods
- Simplex method
- Subgradient method - similar to gradient method in case there are no gradients

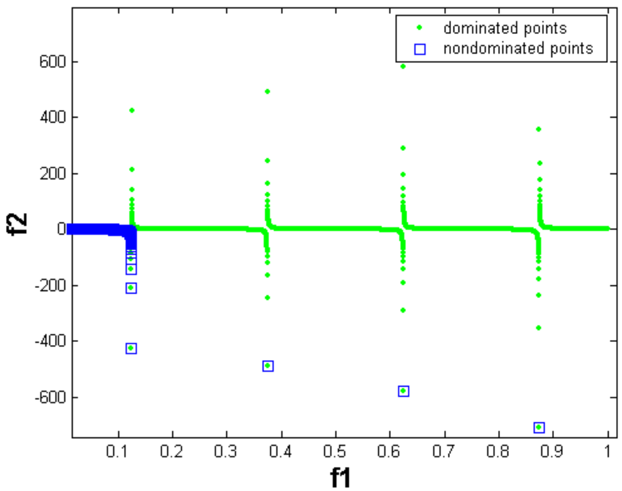
- Constrained problems can often be transformed into unconstrained problems with the help of Lagrange multipliers

Heuristic Approaches

- Ant colony optimization
- Beam search
- Cockroach algorithm
- **Differential Evolution**
- **Evolution Strategy**
- **Evolutionary Programming**
- Firefly algorithm
- **Genetic Algorithm**
- Hill climbing
- **Particle Swarm Optimization**
- ***Simulated annealing***
- Tabu search
- Artificial Immune System
- Bat algorithm
- Bee algorithm
- Cuckoo search
- Dynamic relaxation
- Firework algorithm
- Filled function method
- Harmony search
- IOSO
- Quantum annealing
- Stochastic tunneling

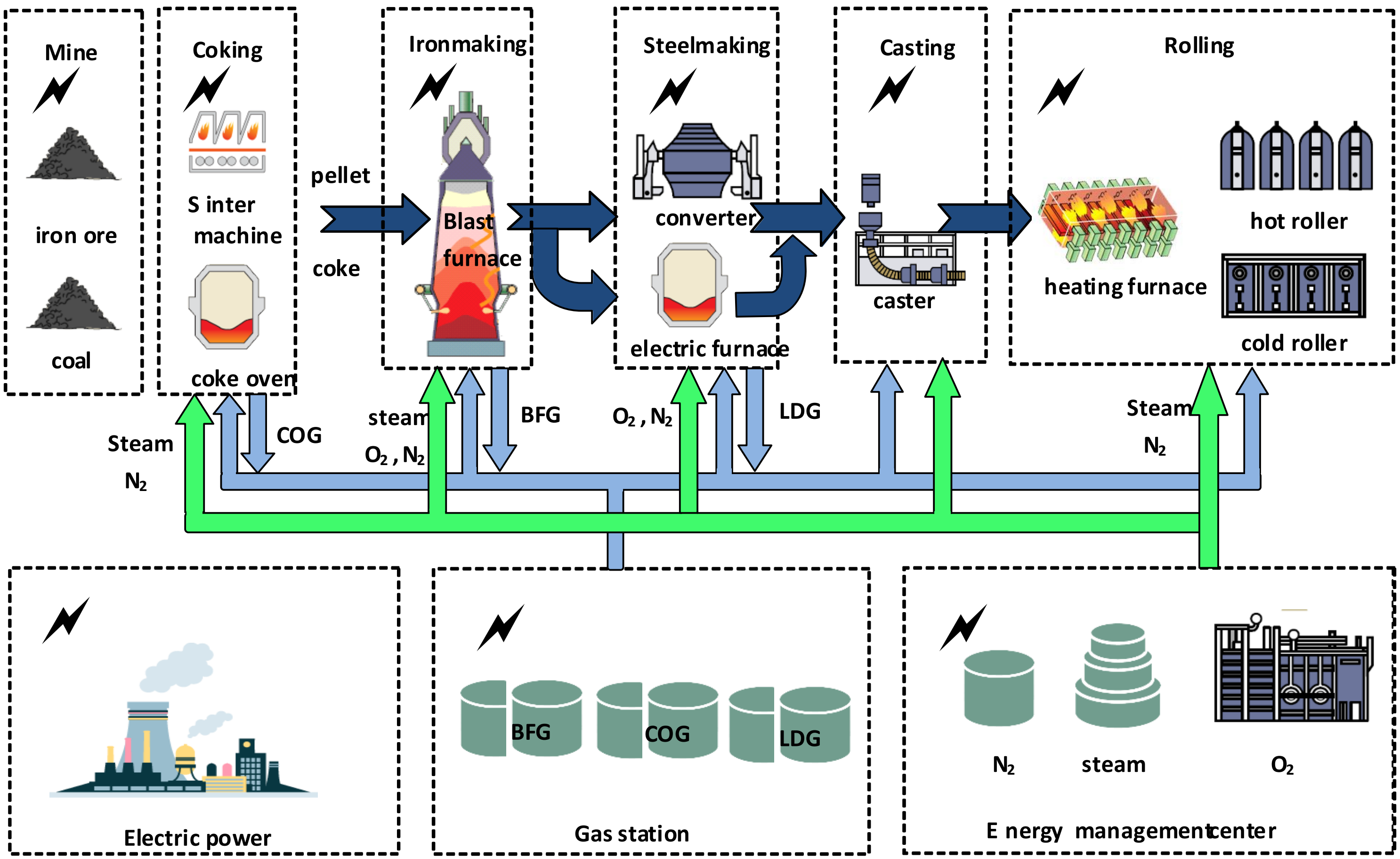
Concerning Issues

- *Existness* of the global optimum
- *Uniqueness* of the global optimum
- *Convergence* of the stochastic algorithm
- *Convergence speed* of the algorithm
- Continuation of the mapping from decision variables to objective functions
- Noise in fitness function and evaluation
- Fuzziness in fitness function and evaluation



Example #1

- *Energy Distribution in an Iron & Steel Factory*



BAOSTEEL

To improve the energy utilization level, the goal of energy distribution problem is to minimize the total energy cost by determining the amount of energy supply for all processes, considering energy supply, holding capacity, regeneration, and conversion.

Objective functions to be minimized:

$$f_1 = \sum_{i=1}^{I} c_0 x_i$$ gas consumption cost

$$f_2 = c_1 z$$ gas purchase cost

$$f_3 = c_2 w$$ emission cost

$$f_4 = \max\{0, h(H + \sum_{i=1}^{I} y_i + z - \sum_{i=1}^{I} x_i - w)\}$$ gas inventory holding cost

Decision variables involved:

x_i	The amount of energy allocated to process i
y_i	The amount of secondary energy generated at process i
z	The purchase of energy
w	The emission of energy

Possible Disturbances:

sensor measurements

model uncertainties

fluctuation of energy market

$$\sum_{i=1}^{I} x_i + w \le \sum_{i} y_i + z + \left(H - H^0\right)$$ gas supply capacity constraint

$$x_i \ge \alpha_i^0 p_i,\ \ i = 1, 2, \dots, I$$ lower bound gas consumption ratio constraint

$$x_i \le \alpha_i^1 p_i,\ \ i = 1, 2, \dots, I$$ upper bound gas consumption ratio constraint

$$H + \sum_{i=1}^{I} y_i + z - \sum_{i=1}^{I} x_i - w \ge H^0$$ lower bound gas pipeline pressure constraint

$$H + \sum_{i=1}^{I} y_i + z - \sum_{i=1}^{I} x_i - w \le H^1$$ upper bound gas pipeline pressure constraint

$$y_i \ge \theta_i^0 p_i,\ \ i = 1, 2, \dots, I$$ lower bound gas regeneration capacity constraint

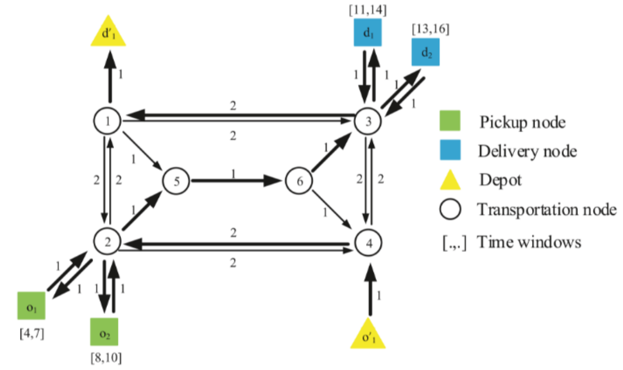
$$y_i \le \theta_i^1 p_i, i = 1, 2, \dots, I$$ upper bound gas regeneration capacity constraint

$$z \le Z, with\ z = \sum_{i=1}^{I} x_i - \sum_{i=1}^{I} y_i - \left(H - H^0\right)$$ additional purchase constraint

$$w \le W, with\ w = \sum_{i=1}^{I} y_i + \left(H^1 - H\right) - \sum_{i=1}^{I} x_i$$ gas emission limit constraint

$$x_i, y_i, z, w \ge 0,\ \ i = 1, 2, \dots, I$$ nonnegative value fields

Example #2



- Pickup and Delivery Problem with Time Windows and Demands (PDP-TW-D)
    - PDP-TW-D is a combinatorial optimization problem of finding a set of optimal routes for a fleet of vehicles in order to serve given transportation requests.
    - Each transportation request is defined by a pickup location, a delivery location, goods to be transported, a pickup time window and a delivery time window.
    - Each pickup/delivery location needs to be visited by a vehicle within a certain time window.
    - Each vehicle has a capacity to load goods and follows an assigned route (i.e., a sequence of pickup/delivery locations) by loading goods at pickup locations and unloading them at delivery locations.
    - Examples: military/civilian airlift and sealift, parcel services, taxi dispatching, shared taxi services, school bus routing, dial-a-ride services, food delivery, and etc.

five conflict objectives:

- the number of vehicles used
- the total travel distance
- the total demands (i.e., the total amount of goods that are transported in a PD plan)
- The total waiting time (i.e., total amountof time that vehicles have to wait if they arrive at nodes before the beginning of their time windows
- the travel distance of the longest route

Three constraints:

- capacity constraint of the vehicle k in a PD route R_k
- time window constraint of the vehicle k in a PD route R_k

Example #3

Problem at hand- which restaurant to go for the dinner?

- Objectives: 1) maximize food quality; 2) maximize portion serving size; 3) minimize driving distance or time
- Decision Variables: 1) cost; 2) reputation/rating from Yelp; 3) business hours; 4) culinary style
- Constraints: 1) cost should be mid-range/affordable for a college student; 2) only interested in 2-star or 3-star; 3) preferable Mexican or Italian food
- Disturbances: 1) possible construction on the shortest route; 2) subjective measure for food quality

Homework #2

<u>Problem 1</u>: Please highlight one multiobjective optimization real-world application, $\bar{y} = \bar{f}(\bar{x})$; specifically identifying 1) the objectives $\bar{y}$ to be minimized/maximized; 2) decision variables $\bar{x}$ involved; 3) possible constraints $\bar{g}$ and $\bar{h}$ and 4) possible disturbances and uncertainities that would imposed upon the objective vector function.

- Due: next Lecture on 6/28/22, before noon
- Please email your answer to TA

Critical Message Conveyed

Real-world optimization problems

tend to be

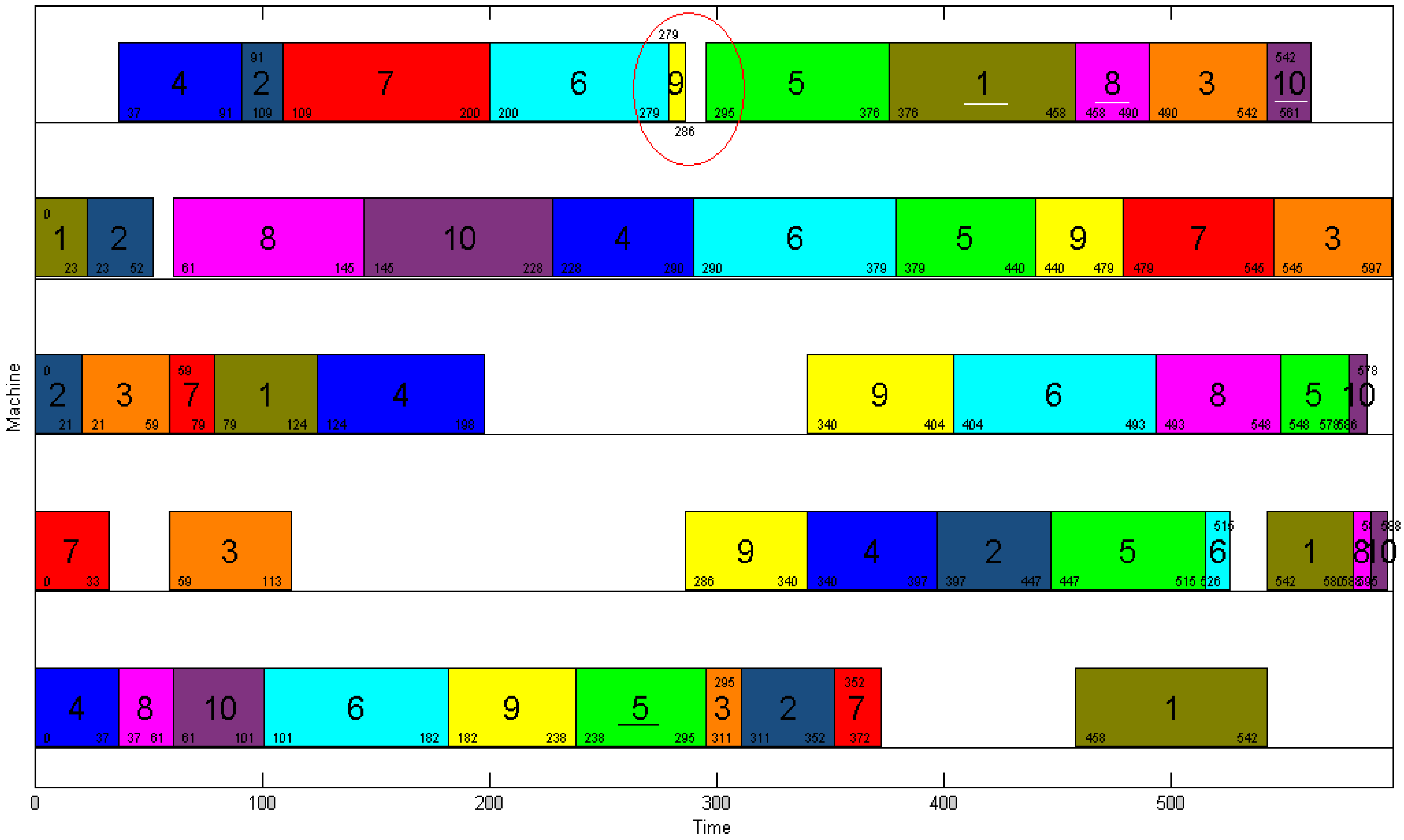
NP-Complete

Including TSPs and JSSs

TSP- 24,978 Cities in Sweden

In May 2004, the traveling salesman problem of visiting all 24,978 cities in Sweden was solved: a tour of length 855,597 TSPLIB units (approximately 72,500 kilometers) was found and it was proven that no shorter tour exists. This is currently the largest solved TSP instance,

JSS- LA03 (10x5)



Q&A