

Evolutionary Computation for Dynamic Multi-objective Optimization Problems

Shengxiang Yang

Centre for Computational Intelligence
School of Computer Science and Informatics
De Montfort University, Leicester LE1 9BH, UK

<http://www.tech.dmu.ac.uk/~syang>

Email: syang@dmu.ac.uk

Tutorial presented at IEEE World Congress on Computational Intelligence (IEEE WCCI 2020), Glasgow, UK, 19-24 July, 2020

Copyright is held by the author



Presenter: Shengxiang Yang

- Education and career history:

- PhD, Northeastern University, China, 1999
- Worked at King's College London, University of Leicester, Brunel University London, 1999-2012
- Joined De Montfort University (DMU) as Professor in Computational Intelligence in July 2012 and
- Director, Centre for Computational Intelligence since 2013



- Research interests:

- Evolutionary Computation (EC) and nature-inspired computation
- Dynamic optimisation and multi-objective optimisation
- Relevant real-world applications
- Over £2M funding

- Over 300 publications with H-index 55 (Google Scholar)

- AE/Editorial Board Member for 10+ journals

- Chaired two IEEE CIS Task Forces

- EC in Dynamic and Uncertain Environments (2011-2017)
- Intelligent Network Systems (2012-2017)

Centre for Computational Intelligence, DMU



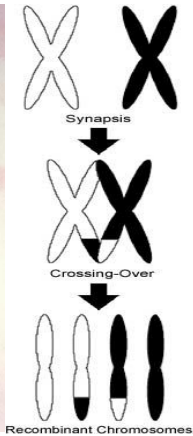
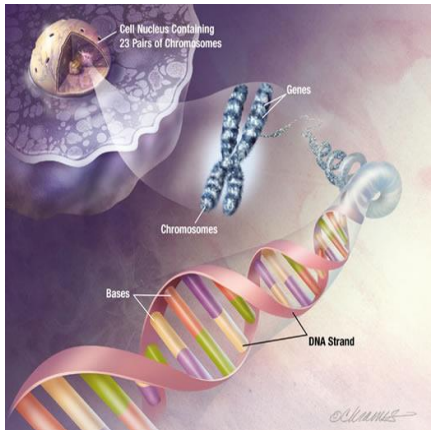
- CCI (www.cci.dmu.ac.uk):
 - **Mission:** Developing fundamental theoretical and practical solutions to real-world problems using a variety of CI paradigms
 - Members: 16 staff, research fellows, ~30 PhDs, **visiting researchers**
 - Themes: EC, fuzzy logic, neural networks, data mining, robotics, game ...
- Funding:
 - Research Councils/Charities: EPSRC, ESRC, EU FP7 & Horizon 2020, Royal Society, Royal Academy of Eng., Innovate UK, KTP, Nuffield Trust ...
 - Government: Leicester City Council, DTI
 - Industries: Lachesis, EMDA, RSSB, Network Rail, etc.
- Collaborations:
 - Universities: UK, USA, Spain, and China
 - Industries and local governments
- Teaching/Training:
 - DTP-IS: University Doctor Training Programme in Intelligent Systems
 - MSc: Intelligent Systems (& Robotics); Data Analytics; BI & Data Mining
 - BSc: AI with Robotics; Computer Game Programming; Math; Comp Sci
- YouTube page: <http://www.youtube.com/thecci>

Outline of the Talk

- Part I: Fundamentals
 - Basic concepts of evolutionary computation (EC)
 - EC for dynamic multi-objective optimization problems (DMOPs): Concept & motivation
 - Classification, benchmarks and test problems
 - Performance measures
- Part II: Approaches, Case Studies, Issues and Future Work
 - EC-based approaches for DMOPs
 - Case studies
 - Challenging issues and future work

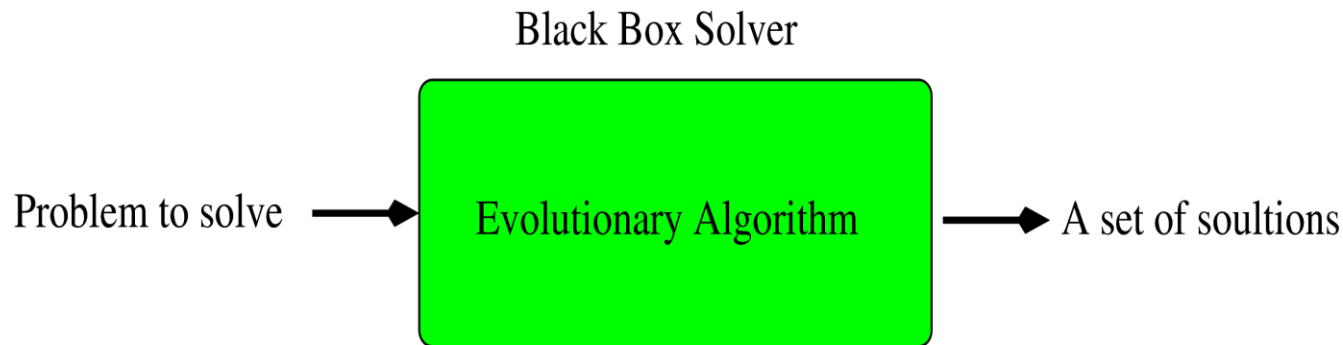
What Is Evolutionary Computation (EC)?

- EC uses mechanisms inspired by
 - **Biological evolution** (e.g., survival of fittest and genetics) or
 - **Biological behaviour** (e.g., ant foraging, bird flocking, animal herding, bacterial growth, fish schooling....)



What Is Evolutionary Computation (EC)?

- EC encapsulates a class of **stochastic optimization algorithms**, dubbed Evolutionary Algorithms (EAs)
- An EA is an **optimisation algorithm** that is
 - **Generic**: a black-box tool for many problems
 - **Population-based**: evolves a population of candidate solutions
 - **Stochastic**: uses probabilistic rules
 - **Bio-inspired**: uses principles inspired from biological evolution or biological behaviour



EC Applications

- Advantages of EC methods:
 - Multiple solutions in a single run
 - No strict requirements to problems
 - Easy to use
- Widely used for optimisation and search problems
 - Financial and economical systems
 - Transportation and logistics systems
 - Industry engineering
 - Automatic programming, art and music design
 -
- EC has also been used for solving multi-objective optimisation problems (MOPs)

Multi-objective Optimization Problems (MOPs)

- MOPs widely exist in real world
- MOPs can be modeled as follows:

$$\text{Min/Max } F(x) = (f_1(x), f_2(x), \dots, f_m(x))$$

$$\text{Subject to } g_j(x) \geq 0, h_k(x) = 0, x^L \leq x \leq x^U$$

- MOPs involve

- Finding vectors of decision variables

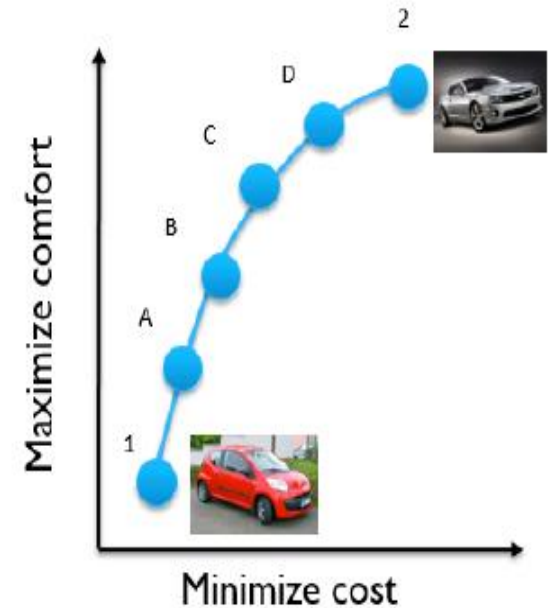
$$x = (x_1, x_2, \dots, x_n) \in S,$$

where S is the **decision space** or **solution space**

- Subject to certain constraints

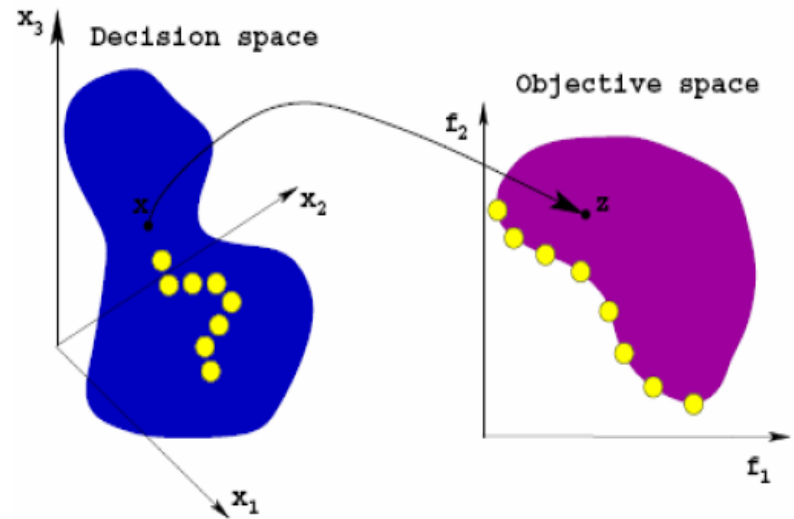
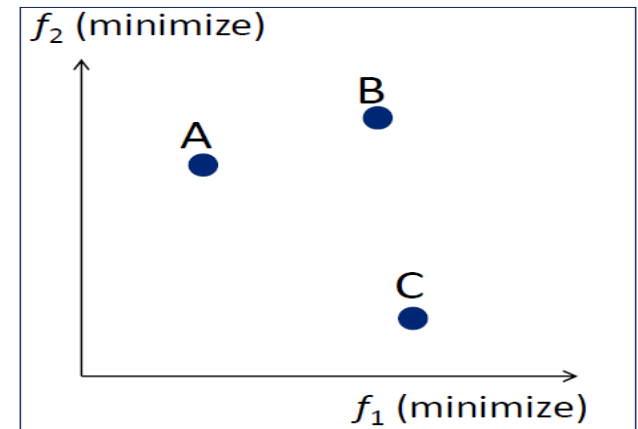
$$x \in F, F \subseteq S, F \text{ contains } \text{feasible solutions}$$

- Simultaneously optimizing m ($m \geq 2$) performance criteria expressed as a vector of (**often conflicting**) objective functions (**objective space**)



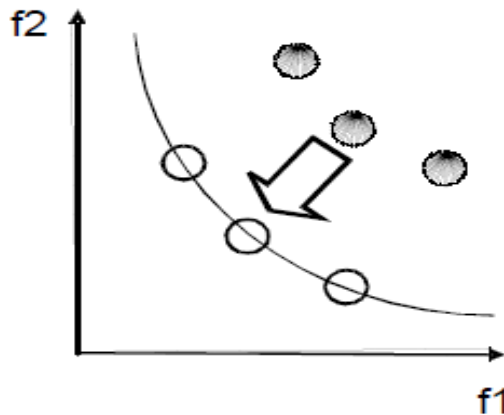
Dominance, Pareto Optimal Set and Front

- Solution x dominates y if & only if
 - x is not worse than y in all objectives
 - x is better than y in at least one objective
- Example:
 - A dominates B, A does not dominate C
- **Pareto optimal set (POS)/Pareto set (PS)**: the set of all non-dominated solutions in the decision space
- **Pareto optimal front (POF)/Pareto front (PF)**: The image of the POS in the objective space

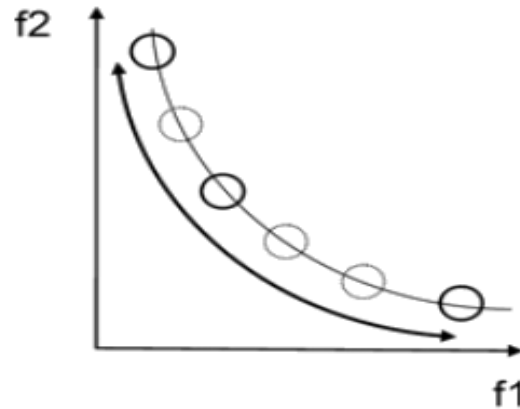


EC for MOPs

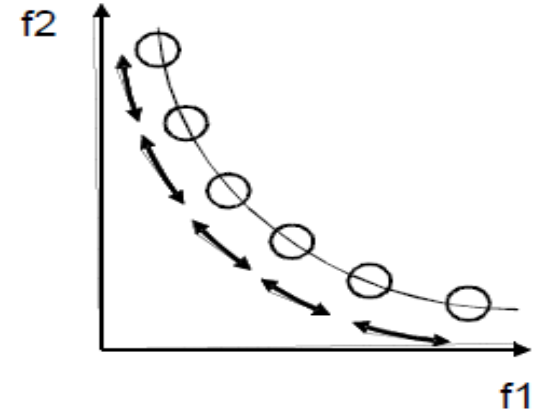
- EC has been used for solving MOPs over four decades
 - Aim to find the PF with three requirements



Close to the Pareto front
(Convergence)



Widely spread
(Extensity)



Equally distributed
(Uniformity)

Diversity: combines extensity and uniformity

- Traditionally, research has focused on static MOPs
- But, many real-world problems are **Dynamic MOPs (DMOPs)**, where changes occur over time
 - In transport networks, travel time between nodes may change
 - In logistics, customer demands may change

What Are DMOPs?

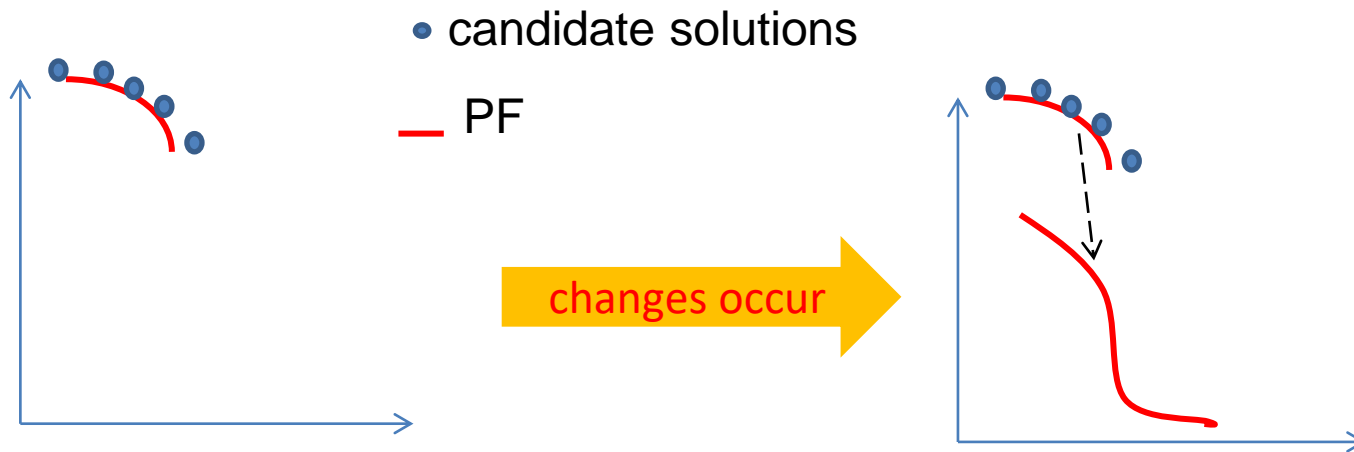
- In general terms, “**optimization problems that involve multiple conflicting objectives and change over time**” are called dynamic or time-dependent multiobjective problems:

$$F = (f_1(x, \varphi, t), f_2(x, \varphi, t), \dots, f_M(x, \varphi, t))^T$$

- x : decision variables;
 - j : parameter;
 - t : time
- A DMOP may involve changes while an algorithm is solving.

Why Are DMOPs Challenging?

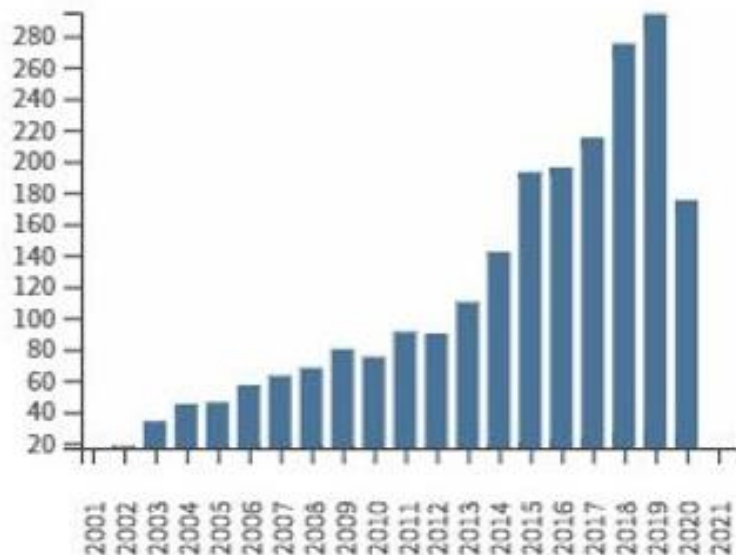
- For DMOPs, PFs and/or PSs may change over time
 - Challenge 1: need to track the moving PF/PS over time
 - Challenge 2: need to re-spread non-dominated solutions



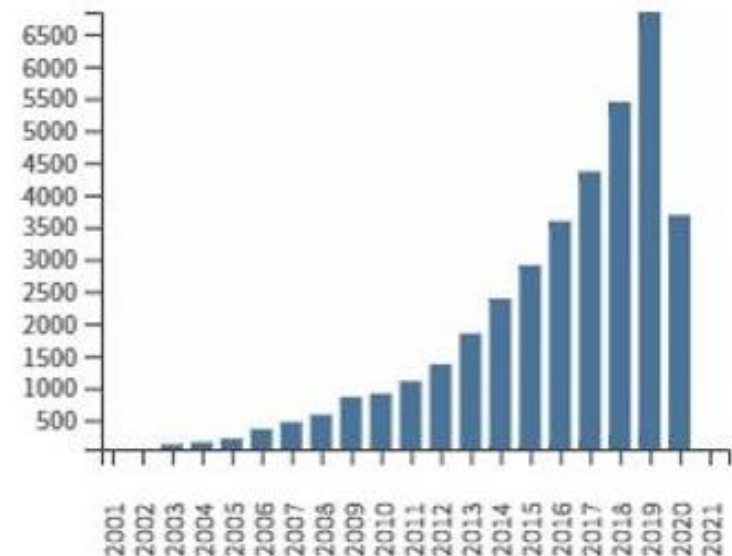
- DMOPs challenge traditional EAs
 - Limited time to respond to environmental changes
 - Once converged, hard to escape from an outdated PF/PS
 - Very likely to lose diversity after a change

Why EC for DMOPs?

- Many real-life problems are DMOPs
 - Desirable to present a set of diverse solutions to decision makers over time
- EAs, once properly modified/enhanced, are good choice
 - Inspired by biological evolution/behaviour, always in dynamic environments
 - Intrinsically, EAs should be good to deal with DMOPs
- Research on EC for DMOPs rises recently
 - Web of Science: TS=((dynamic OR time-varying OR time-dependent OR non-stationary) AND multiobjective AND optimization)



publications by year



citations by year

Classification of DMOPs

- Cause-based rules (*Tantar et al. 2011*):
 - Case 1: the decision variables change over time
 - Case 2: the objective functions change over time
 - Case 3: the current values of decision variables or objective functions depend on their previous values
 - Case 4: parts of or the entire environments change over time
- Effect-based rules (*Farina et al. 2004*):
 - Type I: PS changes, PF remains unchanged
 - Type II: Both PS and PF change
 - Type III: PF changes, PS remains unchanged
 - Type IV: Both PS and PF remain unchanged, although objective functions, constraints, etc., change over time
 - Mixed Type (*Jiang & Yang 2017a*): All of the above four types of change can be present, either randomly or in turn

M. Farina, K. Deb, P. Amato. Dynamic multiobjective optimization problems: test cases, approximations, and applications. *IEEE Transactions on Evolutionary Computation*, 8(5): 425–442, 2004

Benchmark Problems

- Two ideas based on classification rules:
 - Change basic static MOPs to obtain different dynamic effects
 - Introduce novel dynamics that change optimization problems over time
- Real space:
 - Change objective functions with some time-varying factors
 - Dynamically change constraints or the search space
- Combinatorial space:
 - Change decision variables: item weights/profits in multi-objective knapsack problems
 - Add/delete decision variables: nodes added/deleted in network routing problems

Dynamic Multiple Knapsack Problems (DMKPs)

- Static multiple knapsack problems:
 - Given M knapsacks with fixed capacities and n items, each with a weight and profit to each knapsack, select items to fill up the knapsacks to maximize the profit vector while satisfying each knapsack's capacity constraint
- The DMKP (Farina et al. 2004):
 - Constructed by changing weights and profits of items, and/or knapsack capacity over time as:

$$\max f_i(x, t) = \sum_{j=1}^n p_{ij}(t)x_j, \quad i = 1 : M$$

$$s.t. \quad \sum_{j=1}^n w_{ij}(t)x_j \leq c_i(t), \quad i = 1 : M$$

$$x_i \in \{0, 1\}^n$$

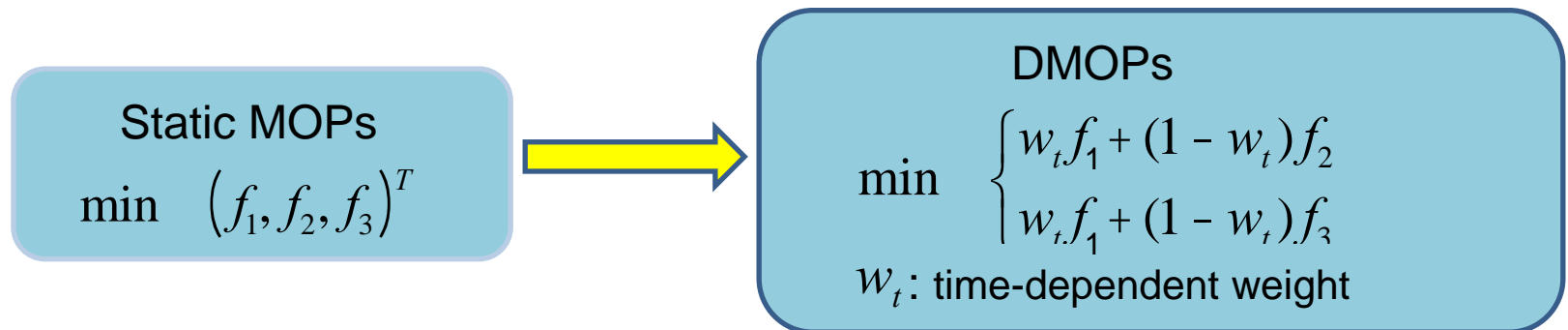
- x_i : indicates whether item i is included or not
- p_{ij} : profit of item i to knapsack j at time t
- w_{ij} : weight of item i to knapsack j at time t
- c_i : the capacity of knapsack i at time t .

Continuous DMOP Benchmarks

- A number of continuous DMOP benchmarks proposed:
 - Jin-Sendhoff's framework by Jin & Sendhoff (2004)
 - FDA test suite by Farina et al. (2004)
 - DSW test problems by Mehnen et al. (2006)
 - dMOP test suite by Goh & Tan (2007)
 - HE test suite by Helbig & Engelbrecht (2013, 2014)
 - UDF test suite by Biswas et al. (2014)
 - F (ZJZ) test suite by Zhou et al. (2014)
 - GTA test suite by Gee et al. (2017)
 - JY generator by Jiang & Yang (2017)

Jin-Sendhoff's Framework (2004)

- Main idea: Aggregating several objective functions with time-varying weights
- For example, a tri-objective minimization problem can be easily transformed into a bi-objective dynamic problem with time-dependent weighted aggregation of any two objectives.



- This framework does not provide well-defined test problems

FDA Test Suite by Farina et al. (2004)

- Consists of 3 ZDT based 2-objective problems and 2 DTLZ based 3-objective problems
- FDA problems (FDA1-FDA3) based on ZDT
 - Problem definition:

$$\min F = (f_1(x, t), g(x, t)h(x, f_1(x, t), g(x, t), t))^T$$

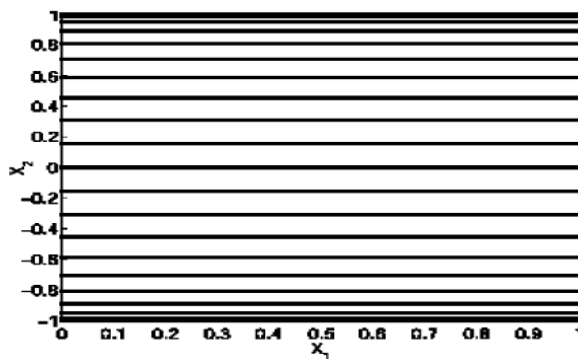
- Scenario 1: time-varying $g(x, t) = 1 + \dot{a}(x_i - G(t))^2$
- Scenario 2: time-varying $h(x, f_1, g, t) = 1 - \left(\frac{f_1}{g}\right)^{(H(t) + \sum (x_i - H(t))^2)^{-1}}$
- Scenario 3: time-varying $f_1(x, t) = \dot{a}x_i^{F(t)}$, $F(t) > 0$
- Scenario 4: change g, h, and f1 functions simultaneously

FDA Test Suite by Farina et al. (2004) - 2

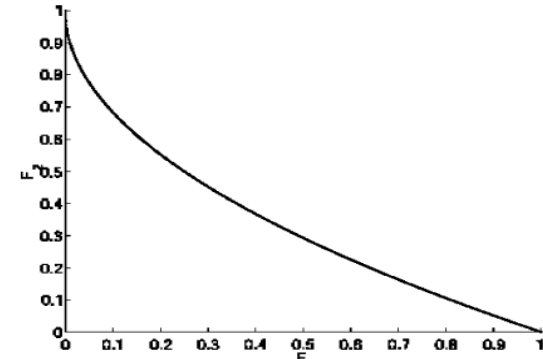
FDA1 (Type I)

$$\left\{ \begin{array}{l} f_1(\mathbf{x}_I) = x_1 \\ g(\mathbf{x}_{II}) = 1 + \sum_{x_i \in \mathbf{x}_{II}} (x_i - G(t))^2 \\ h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} \\ G(t) = \sin(0.5\pi t), \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_T} \right\rfloor \\ \mathbf{x}_I = (x_1) \in [0, 1], \quad \mathbf{x}_{II} = (x_2, \dots, x_n) \in [-1, 1] \end{array} \right.$$

PS



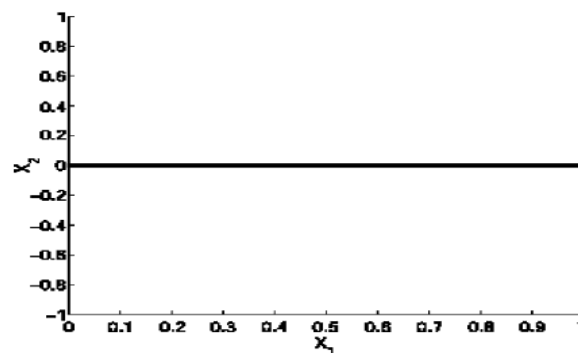
PF



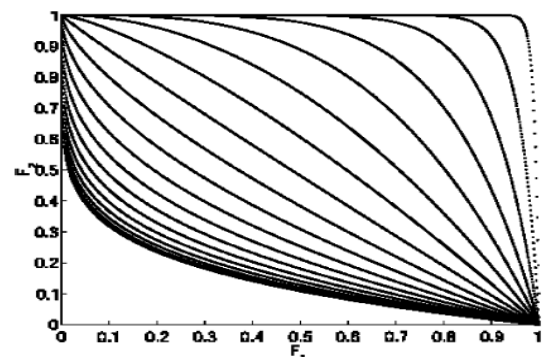
FDA2 (Type III)

$$\left\{ \begin{array}{l} f_1(\mathbf{x}_I) = x_1 \\ g(\mathbf{x}_{II}) = 1 + \sum_{x_i \in \mathbf{x}_{II}} x_i^2 \\ h(\mathbf{x}_{III}, f_1, g) = 1 - \left(\frac{f_1}{g}\right) \left(H(t) + \sum_{x_i \in \mathbf{x}_{III}} (x_i - H(t))^2\right)^{-1} \\ H(t) = 0.75 + 0.7 \sin(0.5\pi t), \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_T} \right\rfloor \\ \mathbf{x}_I = (x_1) \in [0, 1], \quad \mathbf{x}_{II}, \mathbf{x}_{III} \in [-1, 1] \end{array} \right.$$

PS



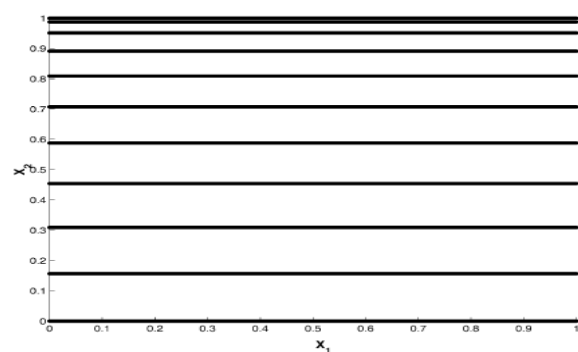
PF



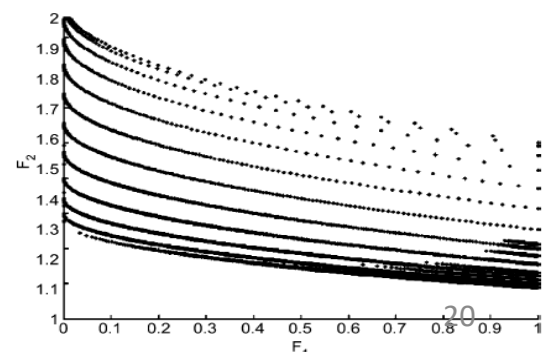
FDA3 (Type II)

$$\left\{ \begin{array}{l} f_1(\mathbf{x}_I) = \sum_{x_i \in \mathbf{x}_I} x_i^{F(t)} \\ g(\mathbf{x}_{II}) = 1 + G(t) + \sum_{x_i \in \mathbf{x}_{II}} (x_i - G(t))^2 \\ h(f_1, g) = 1 - \sqrt{\frac{f_1}{g}} \\ G(t) = |\sin(0.5\pi t)| \\ F(t) = 10^{2 \sin(0.5\pi t)}, \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_T} \right\rfloor \\ \mathbf{x}_I \in [0, 1] \quad \mathbf{x}_{II} \in [-1, 1] \end{array} \right.$$

PS



PF



FDA Test Suite by Farina et al. (2004) - 3

- FDA problems (FDA4-FDA5) based on DTLZ

- Problem definition:

$$\text{Min. } f_1(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos\left(\frac{x_1\pi}{2}\right) \cdots \cos\left(\frac{x_{M-1}\pi}{2}\right)$$

$$\text{Min. } f_2(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \cos\left(\frac{x_1\pi}{2}\right) \cdots \sin\left(\frac{x_{M-1}\pi}{2}\right)$$

⋮

$$\text{Min. } f_M(\mathbf{x}) = (1 + g(\mathbf{x}_M)) \sin\left(\frac{x_1\pi}{2}\right)$$

$$\text{with } g(\mathbf{x}_M) = \sum_{x_i \in \mathbf{x}_M} (x_i - 0.5)^2$$
$$0 \leq x_i \leq 1, \text{ for } i = 1, 2, \dots, n$$

- Scenario 1: the change of $g(x_M, t) = G(t) + \sum (x_i - G(t))^2, G(t) > 0$

- Scenario 2: the change of $x_i \rightarrow x_i^{F(x)}$

- Scenario 3: PF shape variation, the change of $g(x, t) \rightarrow g(x, t) + K_i(t)$ for each objective function

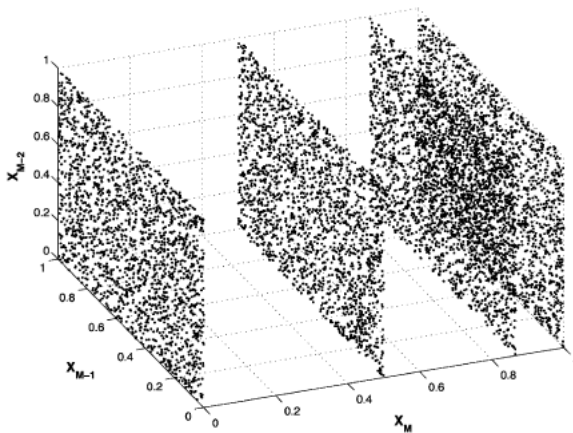
FDA Test Suite by Farina et al. (2004) - 4

FDA4 (Type I)

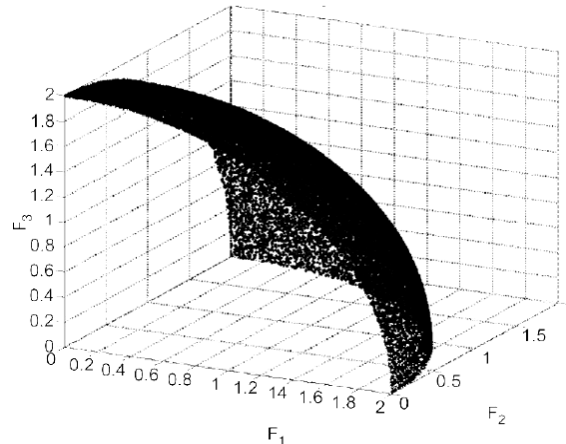
$$\left\{ \begin{array}{l} \min_{\mathbf{x}} \quad f_1(\mathbf{x}) = (1 + g(\mathbf{x}_{II})) \prod_{i=1}^{M-1} \cos\left(\frac{x_i \pi}{2}\right) \\ \min_{\mathbf{x}} \quad f_k(\mathbf{x}) = (1 + g(\mathbf{x}_{II})) \left(\prod_{i=1}^{M-k} \cos\left(\frac{x_i \pi}{2}\right) \right) \\ \quad \sin\left(\frac{x_{M-k+1} \pi}{2}\right), \quad k = 2 : M - 1 \\ \min_{\mathbf{x}} \quad f_M(\mathbf{x}) = (1 + g(\mathbf{x}_{II})) \sin\left(\frac{x_1 \pi}{2}\right) \\ \text{where} \quad g(\mathbf{x}_{II}) = \sum_{x_i \in \mathbf{x}_{II}} (x_i - G(t))^2 \\ \quad G(t) = |\sin(0.5\pi t)|, \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_T} \right\rfloor \\ \quad \mathbf{x}_{II} = (x_M, \dots, x_n), \\ \quad x_i \in [0, 1] \quad i = 1 : n \end{array} \right.$$

FDA5 (Type II)

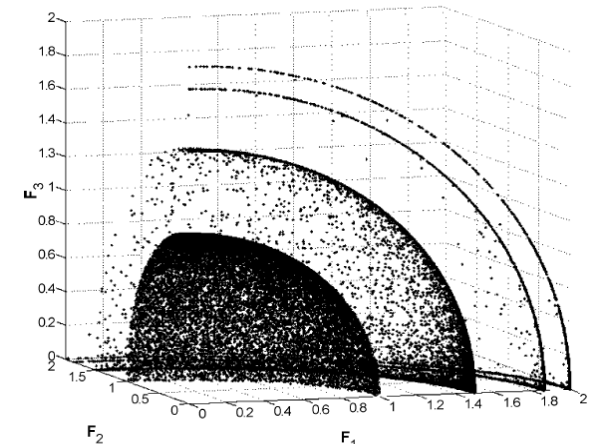
$$\left\{ \begin{array}{l} \min_{\mathbf{x}} \quad f_1(\mathbf{x}) = (1 + g(\mathbf{x}_{II})) \prod_{i=1}^{M-1} \cos\left(\frac{y_i \pi}{2}\right) \\ \min_{\mathbf{x}} \quad f_k(\mathbf{x}) = (1 + g(\mathbf{x}_{II})) \left(\prod_{i=1}^{M-k} \cos\left(\frac{y_i \pi}{2}\right) \right) \\ \quad \sin\left(\frac{y_{M-k+1} \pi}{2}\right) \quad k = 2 : M - 1 \\ \min_{\mathbf{x}} \quad f_M(\mathbf{x}) = (1 + g(\mathbf{x}_{II})) \sin\left(\frac{y_1 \pi}{2}\right) \\ \text{where} \quad g(\mathbf{x}_{II}) = G(t) + \sum_{x_i \in \mathbf{x}_{II}} (x_i - G(t))^2 \\ \quad y_i = x_i^{F(t)} \quad \text{for } i = 1, \dots, (M - 1) \\ \quad G(t) = |\sin(0.5\pi t)| \\ \quad F(t) = 1 + 100 \sin^4(0.5\pi t) \\ \quad t = \frac{1}{n_t} \left\lfloor \frac{\tau}{\tau_T} \right\rfloor \\ \quad \mathbf{x}_{II} = (x_M, \dots, x_n), \quad x_i \in [0, 1], \quad i = 1 : n \end{array} \right.$$



PS of FDA4 & FDA5



PF of FDA4



PF of FDA5

DSW Test Problems by Mehnen et al. (2006)

- Motivated by single-objective unimodal sphere models

$$\text{DSW} = \begin{cases} \text{Minimise: } f(\mathbf{x}, t) = (f_1(\mathbf{x}, t), f_2(\mathbf{x}, t)) \\ f_1(\mathbf{x}, t) = (a_{11}x_1 + a_{12}|x_1| - b_1G(t))^2 + \sum_{i=2}^n x_i^2 \\ f_2(\mathbf{x}, t) = (a_{21}x_1 + a_{22}|x_1| - b_2G(t) - 2)^2 + \sum_{i=2}^n x_i^2 \\ \text{where: } G(t) = t(\tau)s, t(\tau) = \frac{1}{n_t} \lfloor \frac{\tau}{\tau_t} \rfloor \\ \quad \quad \quad s \text{ representing the severity of change} \end{cases}$$

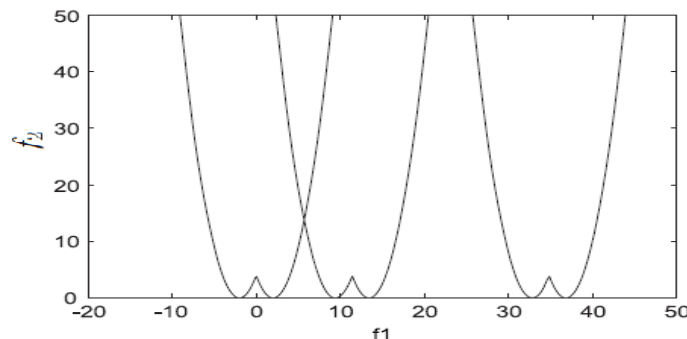
- Three cases generated by varying a, b parameters

- DSW1: time-changing continuous PS bounds
- DSW2: time-changing disconnected PS
- DSW3: time-changing PS and PF

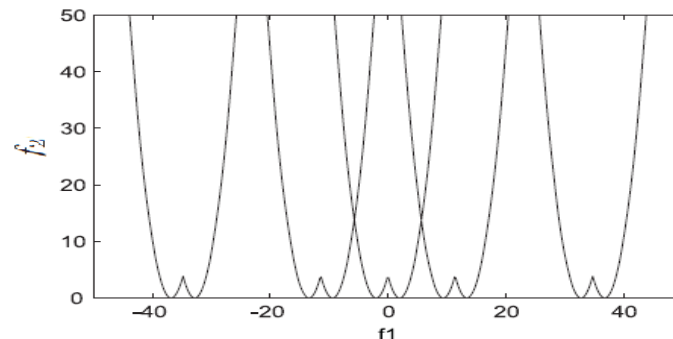
DSW1: $\mathbf{x} \in [-50, 50]^n, a_{11} = 1, a_{12} = 0, a_{21} = 1, a_{21} = 0, b_1 = b_2 = 1$

DSW2: $\mathbf{x} \in [-50, 50]^n, a_{11} = 0, a_{12} = 1, a_{21} = 0, a_{21} = 1, b_1 = b_2 = 1$

DSW3: $\mathbf{x} \in [-50, 50]^n, a_{11} = 1, a_{12} = 0, a_{21} = 1, a_{21} = 0, b_1 = 0, b_2 = 1$



POF of DSW1



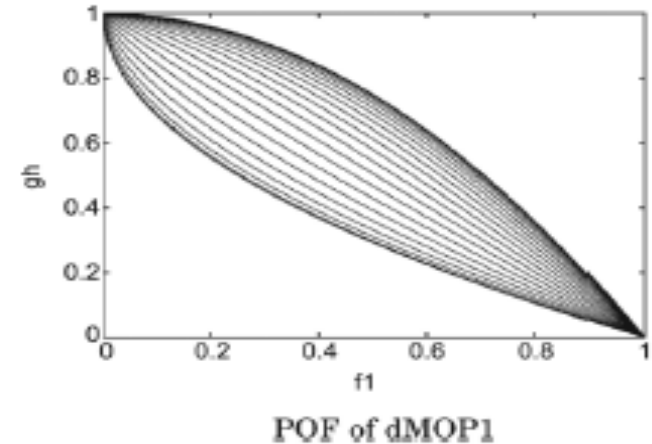
POF of DSW2

dMOP Test Suite by Goh & Tan (2007)

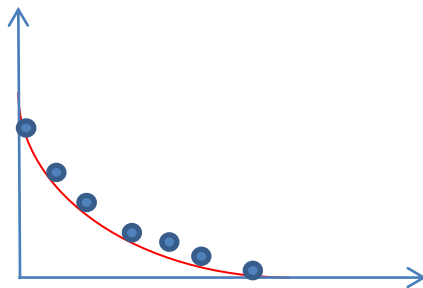
- Derived from ZDT problems and FDA problems
- Three bi-objective problems

➤ dMOP1-2 similar to FDA

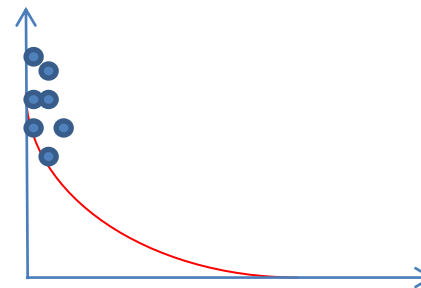
$$\text{dMOP1} = \begin{cases} \text{Minimise: } f(\mathbf{x}, t) = (f_1(\mathbf{x}_I), g(\mathbf{x}_{II}) \cdot h(f_1(\mathbf{x}_I), g(\mathbf{x}_{II}), t)) \\ f_1(\mathbf{x}_I) = x_1 \\ g(\mathbf{x}_{II}) = 1 + 9 \sum_{x_i \in \mathbf{x}_{II}} (x_i)^2 \\ h(f_1, g, t) = 1 - \left(\frac{f_1}{g}\right)^{H(t)} \\ \text{where:} \\ H(t) = 0.75 \sin(0.5\pi t) + 1.25, \quad t = \frac{1}{n_t} \lfloor \frac{\tau}{\tau_t} \rfloor \\ x_i \in [0, 1]; \quad \mathbf{x}_I = (x_1); \quad \mathbf{x}_{II} = (x_2, \dots, x_n) \end{cases}$$



➤ dMOP3 involves randomness and marked diversity loss



objective space,
population at time t



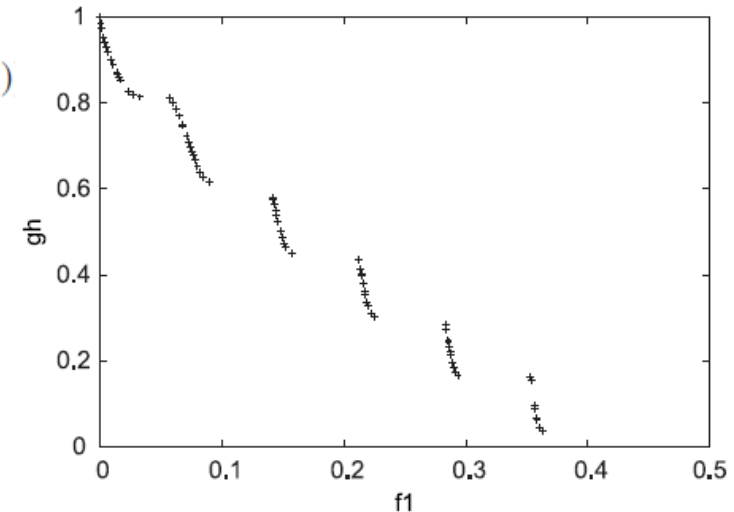
objective space,
population at time t+1

HE Test Suite by Helbig & Engelbrecht (2013)

- Adding WFG (Huband et al. 2006) characteristics:

- isolated PFs
- deceptive PFs
- ...

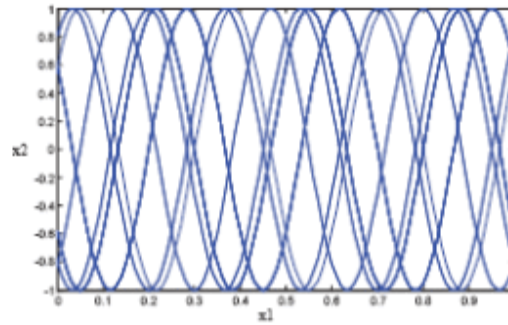
$$\text{HE1} = \begin{cases} \text{Minimise: } f(\mathbf{x}, t) = (f_1(\mathbf{x}_I), g(\mathbf{x}_{II}) \cdot h(f_1(\mathbf{x}_I), g(\mathbf{x}_{II}), t)) \\ f_1(\mathbf{x}_I) = x_1 \\ g(\mathbf{x}_{II}) = 1 + \frac{9}{n-1} \sum_{x_i \in \mathbf{x}_{II}} x_i \\ h(f_1, g, t) = 1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g} \sin(10\pi t f_1) \\ \text{where:} \\ t = \frac{1}{n_t} \lfloor \frac{\tau}{\tau_t} \rfloor \\ x_i \in [0, 1]; \mathbf{x}_I = (x_1); \mathbf{x}_{II} = (x_2, \dots, x_n) \end{cases}$$



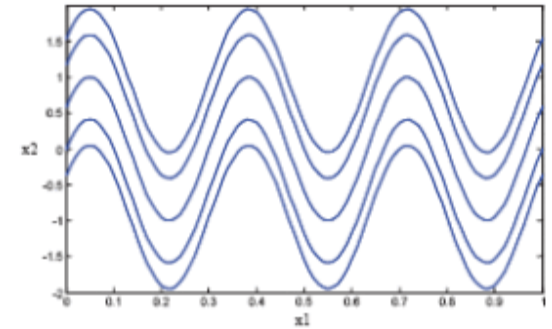
- However, main optimization difficulties come from WFG characteristics instead of introduced dynamics

UDF Test Suite by Biswas et al. (2014)

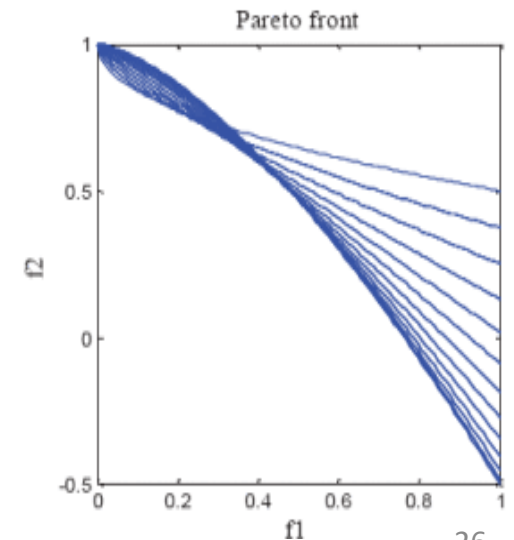
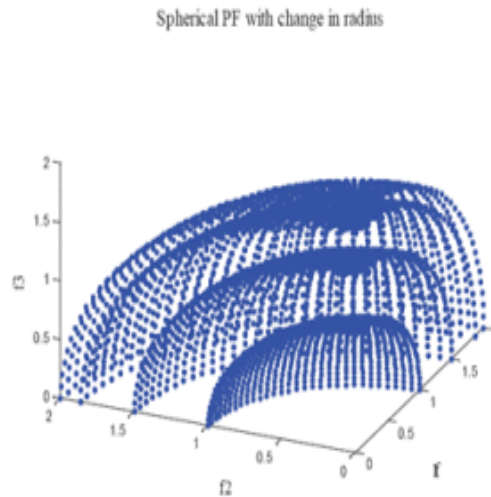
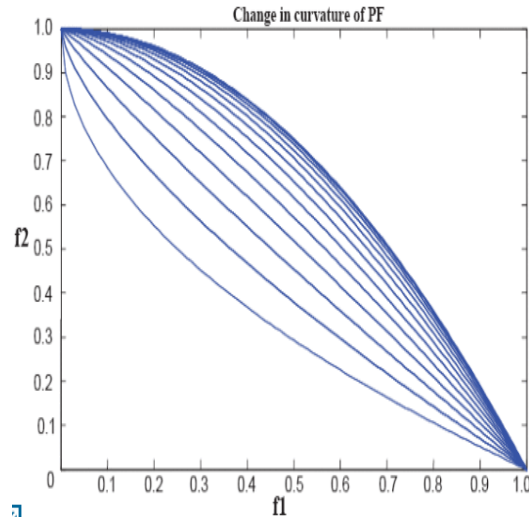
- Based on UF problems (Zhang et al. 2009)
- General techniques to design DMOPs
 - Shifting
 - Shape variation
 - Slope variation
 - Curvature variation
 - ...



(a) Horizontal Shift of the PS in 2D



(b) Vertical Shift of the PS in 2D



The F (ZJZ) Test Suite by Zhou et al. (2014)

- Based on UF test problems (Zhang et al. 2009)
- F1-F4 are the same as FDA1-FDA4 (Farina et al. 2004)
- Involving strong nonlinear variable linkages

F5 $[0, 5]^n$

$$f_1(x, t) = |x_1 - a|^H + \sum_{i \in I_1} y_i^2,$$

$$f_2(x, t) = |x_1 - a - 1|^H + \sum_{i \in I_2} y_i^2,$$

$$y_i = x_i - b - 1 + |x_1 - a|^{H + \frac{i}{n}}, H = 1.25 + 0.75 \sin(\pi \frac{t}{nT}),$$

$$a = 2 \cos(\pi \frac{t}{nT}) + 2, b = 2 \sin(2\pi \frac{t}{nT}) + 2,$$

$$I_1 = \{i | 1 \leq i \leq n, i \text{ is odd}\}, I_2 = \{i | 1 \leq i \leq n, i \text{ is even}\}.$$

$$\text{PS}(t): a \leq x_1 \leq a + 1, x_i = b + 1 - |x_1 - a|^{H + \frac{i}{n}}, \text{ for } i = 2, \dots, n.$$

$$\text{PF}(t): f_1 = s^H, f_2 = (1 - s)^H, 0 \leq s \leq 1.$$

F6 $[0, 5]^n$

$$f_1(x, t) = |x_1 - a|^H + \sum_{i \in I_1} y_i^2,$$

$$f_2(x, t) = |x_1 - a - 1|^H + \sum_{i \in I_2} y_i^2,$$

$$y_i = x_i - b - 1 + |x_1 - a|^{H + \frac{i}{n}}, H = 1.25 + 0.75 \sin(\pi \frac{t}{nT}),$$

$$a = 2 \cos(1.5\pi \frac{t}{nT}) \sin(0.5\pi \frac{t}{nT}) + 2, b = 2 \cos(1.5\pi \frac{t}{nT}) \cos(0.5\pi \frac{t}{nT}) + 2$$

$$I_1 = \{i | 1 \leq i \leq n, i \text{ is odd}\}, I_2 = \{i | 1 \leq i \leq n, i \text{ is even}\}.$$

$$\text{PS}(t): a \leq x_1 \leq a + 1, x_i = b + 1 - |x_1 - a|^{H + \frac{i}{n}}, \text{ for } i = 2, \dots, n.$$

$$\text{PF}(t): f_1 = s^H, f_2 = (1 - s)^H, 0 \leq s \leq 1.$$

F7 $[0, 5]^n$

$$f_1(x, t) = |x_1 - a|^H + \sum_{i \in I_1} y_i^2,$$

$$f_2(x, t) = |x_1 - a - 1|^H + \sum_{i \in I_2} y_i^2,$$

$$y_i = x_i - b - 1 + |x_1 - a|^{H + \frac{i}{n}}, H = 1.25 + 0.75 \sin(\pi \frac{t}{nT}),$$

$$a = 1.7(1 - \sin(\pi \frac{t}{nT})) \sin(\pi \frac{t}{nT}) + 3.4, b = 1.4(1 - \sin(\pi \frac{t}{nT})) \cos(\pi \frac{t}{nT}) + 2.1$$

$$I_1 = \{i | 1 \leq i \leq n, i \text{ is odd}\}, I_2 = \{i | 1 \leq i \leq n, i \text{ is even}\}.$$

$$\text{PS}(t): a \leq x_1 \leq a + 1, x_i = b + 1 - |x_1 - a|^{H + \frac{i}{n}}, \text{ for } i = 2, \dots, n.$$

$$\text{PF}(t): f_1 = s^H, f_2 = (1 - s)^H, 0 \leq s \leq 1.$$

F8 $[0, 1]^2 \times [-1, 2]^{n-2}$

$$f_1(x, t) = (1 + g) \cos(0.5\pi x_2) \cos(0.5\pi x_1),$$

$$f_2(x, t) = (1 + g) \cos(0.5\pi x_2) \sin(0.5\pi x_1),$$

$$f_3(x, t) = (1 + g) \sin(0.5\pi x_2),$$

$$g = \sum_{i=3}^n (x_i - (\frac{x_1 + x_2}{2})^H - G)^2,$$

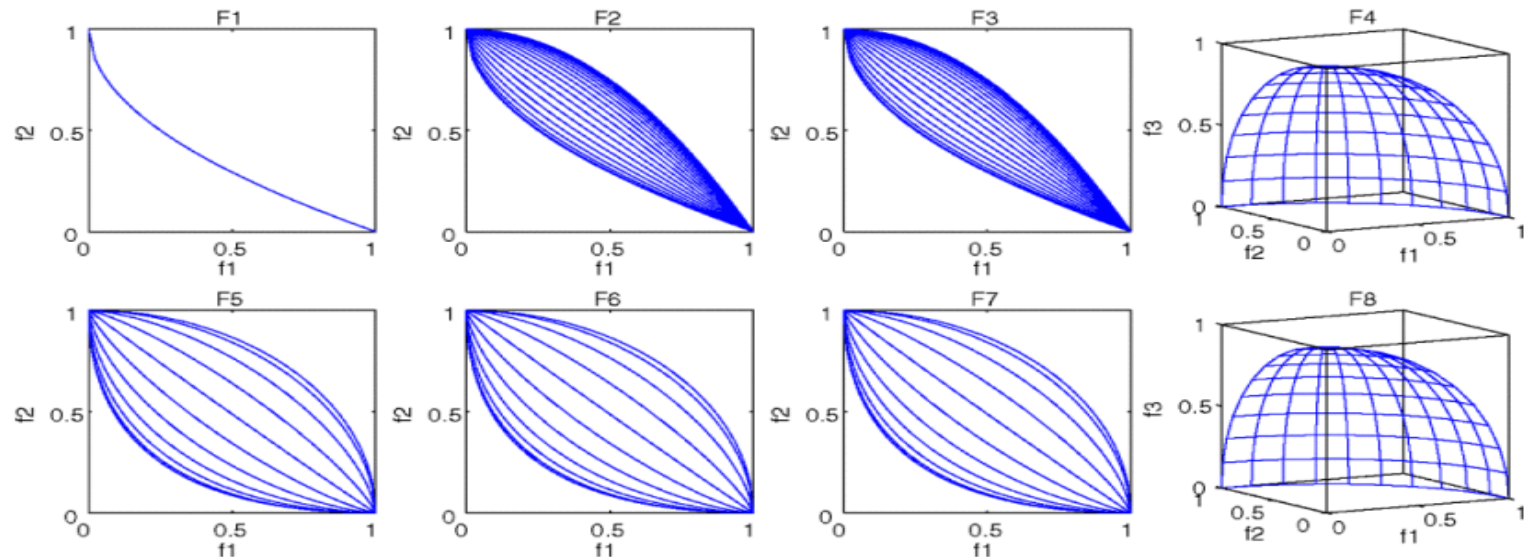
$$H = 1.25 + 0.75 \sin(\pi \frac{t}{nT}), G = \sin(0.5\pi \frac{t}{nT}).$$

$$\text{PS}(t): 0 \leq x_1, x_2 \leq 1, x_i = (\frac{x_1 + x_2}{2})^H + G(t), \text{ for } i = 3, \dots, n.$$

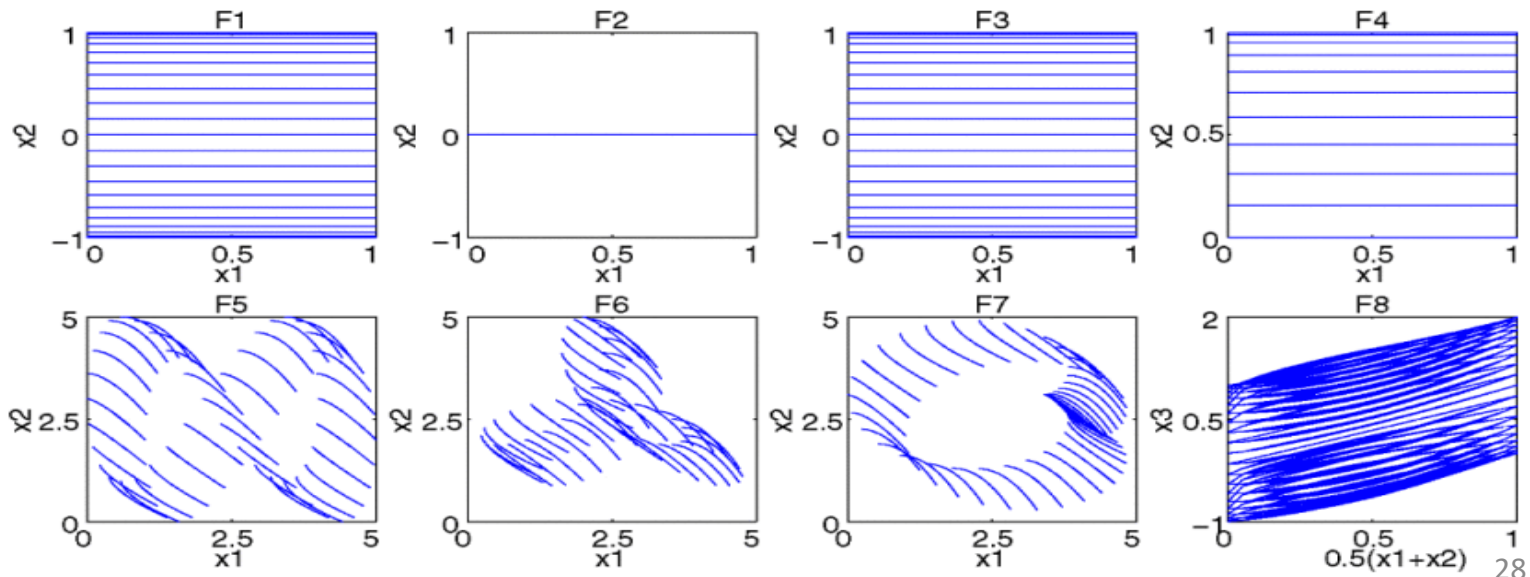
$$\text{PF}(t): f_1 = \cos(u) \cos(v), f_2 = \cos(u) \sin(v), f_3 = \sin(u), 0 \leq u, v \leq \pi/2.$$

The F (ZJZ) Test Suite by Zhou et al. (2014)

PF:



PS:



JY Generator by Jiang & Yang (2017)

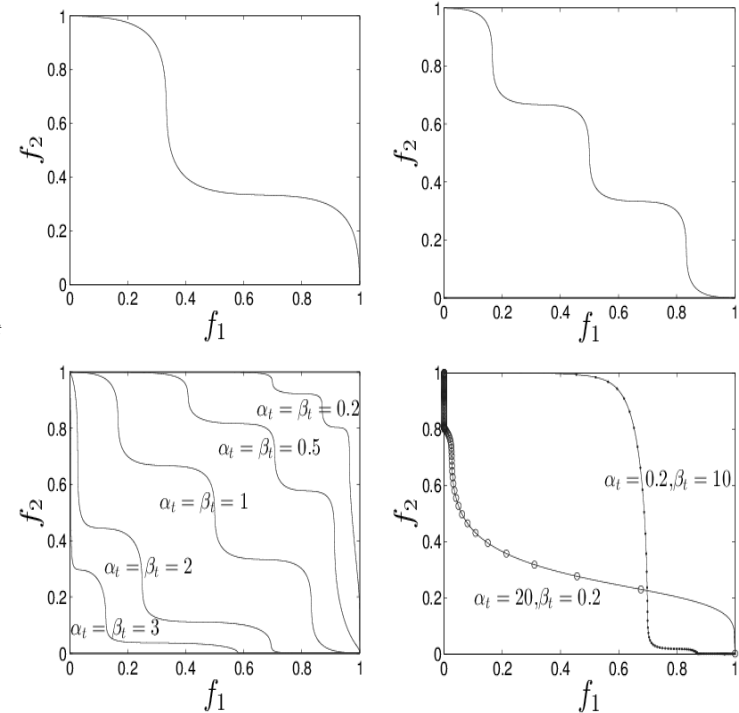
- Focusing on dynamics analysis

$$JY = \begin{cases} \min & (f_1(x,t), f_2(x,t))^T \\ f_1(x,t) & = (1 + g(x,t))(h(x) + A_t \sin(W_t \pi h(x)))^{\alpha_t} \\ f_2(x,t) & = (1 + g(x,t))(1 - h(x) + A_t \sin(W_t \pi h(x)))^{\beta_t} \end{cases}$$

$$\text{PF: } f_1^{\frac{1}{\alpha_t}} + f_1^{\frac{1}{\beta_t}} = 1 + 2A_t \sin\left(W_t \pi \frac{f_1^{\frac{1}{\alpha_t}} - f_1^{\frac{1}{\beta_t}} + 1}{2}\right)$$

- Characteristics:

- PF is a sin wave after a clockwise rotation
- The PF has mixed concave and convex segments
- Time-varying segments controlled by W_t
- Time-varying curvature controlled by A_t
- Various types of problems, e.g., randomness, knee regions, dis-connectivity
- Easy to scale up in terms of the number of objectives

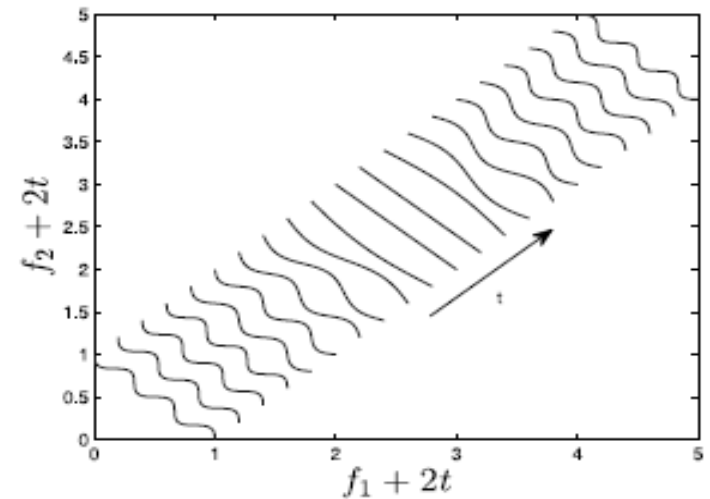


JY Generator by Jiang & Yang (2017) - 2

- JY2: time-changing PS and PF

$$JY2 : \begin{cases} \min F(\mathbf{x}, t) = (f_1(\mathbf{x}, t), f_2(\mathbf{x}, t))^T \\ f_1(\mathbf{x}, t) = (1 + g(\mathbf{x}_{II}, t))(x_1 + A_t \sin(W_t \pi x_1)) \\ f_2(\mathbf{x}, t) = (1 + g(\mathbf{x}_{II}, t))(1 - x_1 + A_t \sin(W_t \pi x_1)) \\ g(\mathbf{x}_{II}, t) = \sum_{x_i \in \mathbf{x}_{II}} (x_i - G(t))^2, G(t) = \sin(0.5\pi t) \\ A(t) = 0.05, W(t) = \lfloor 6\sin(0.5\pi(t - 1)) \rfloor \\ \mathbf{x}_I = (x_1) \in [0, 1], \mathbf{x}_{II} = (x_2, \dots, x_n) \in [-1, 1]^{n-1} \end{cases}$$

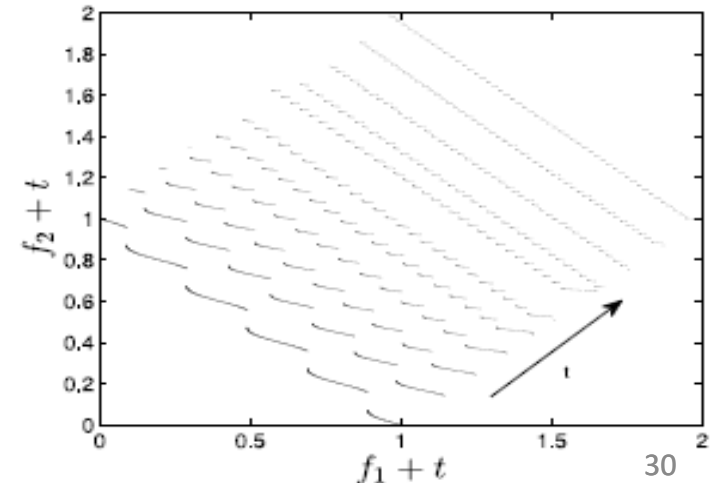
POF of JY2 with 21 time windows varying from 0 to 2.



- JY4: time-changing PS and PF, time-changing disconnectivity

$$JY4 : \begin{cases} \min F(\mathbf{x}, t) = (f_1(\mathbf{x}, t), f_2(\mathbf{x}, t))^T \\ f_1(\mathbf{x}, t) = (1 + g(\mathbf{x}_{II}, t))(x_1 + A_t \sin(W_t \pi x_1)) \\ f_2(\mathbf{x}, t) = (1 + g(\mathbf{x}_{II}, t))(1 - x_1 + A_t \sin(W_t \pi x_1)) \\ g(\mathbf{x}_{II}, t) = \sum_{x_i \in \mathbf{x}_{II}} (x_i - G(t))^2, G(t) = \sin(0.5\pi t) \\ A(t) = 0.05, W(t) = \lfloor 10^{1+|G(t)|} \rfloor \\ \mathbf{x}_I = (x_1) \in [0, 1], \mathbf{x}_{II} = (x_2, \dots, x_n) \in [-1, 1]^{n-1} \end{cases}$$

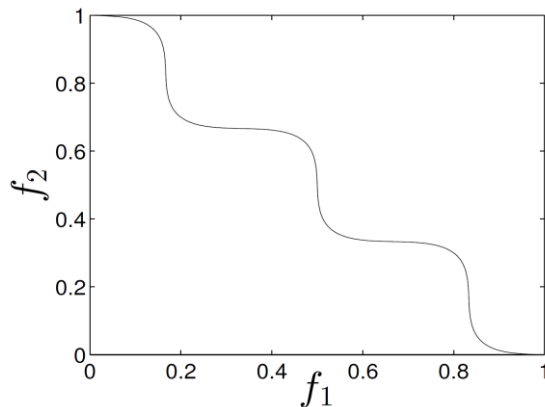
POF of JY4 with 11 time windows varying from 0 to 2.



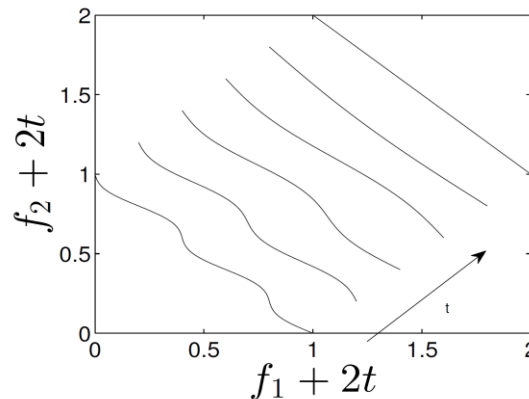
JY Generator by Jiang & Yang (2017) - 3

- JY10: **mixed type**, sometimes PS remains static whereas sometimes PS changes over time. PF has the same dynamics

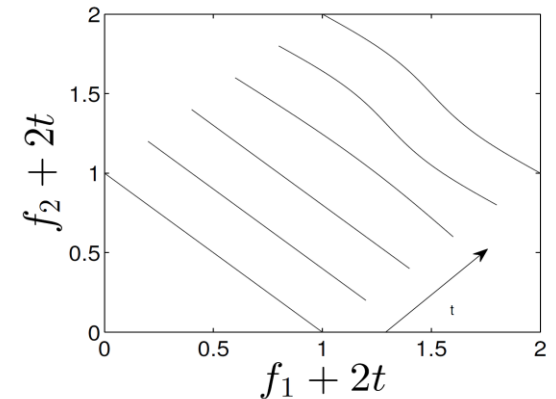
$$\text{JY10: } \left\{ \begin{array}{l} \min \quad F(\mathbf{x}, t) = (f_1(\mathbf{x}, t), f_2(\mathbf{x}, t))^T \\ f_1(\mathbf{x}, t) = (1 + g(\mathbf{x}_{\mathbf{II}}, t))(x_1 + A_t \sin(W_t \pi x_1))^{\alpha_t} \\ f_2(\mathbf{x}, t) = (1 + g(\mathbf{x}_{\mathbf{II}}, t))(1 - x_1 + A_t \sin(W_t \pi x_1))^{\beta_t} \\ g(\mathbf{x}_{\mathbf{II}}, t) = \sum_{x_i \in \mathbf{x}_{\mathbf{II}}} (x_i + \sigma - G(t))^2, G(t) = |\sin(0.5\pi t)| \\ A(t) = 0.05, \quad W(t) = 6 \\ \alpha_t = \beta_t = 1 + \sigma G(t), \sigma \equiv (\lfloor \frac{\tau}{\tau_i \rho_i} \rfloor + R) \pmod{3} \\ \mathbf{x}_{\mathbf{I}} = (x_1) \in [0, 1], \mathbf{x}_{\mathbf{II}} = (x_2, \dots, x_n) \in [-1, 1]^{n-1}, \end{array} \right.$$



Static PF, dynamic PS



Dynamic PF, dynamic PS



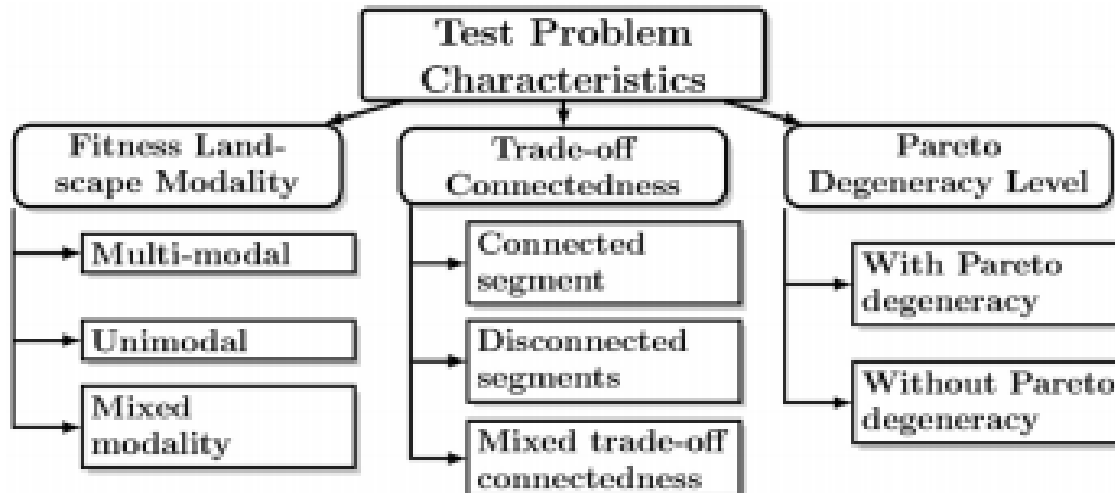
Dynamic PF, static PS

GTA Test Suite by Gee et al. (2017)

- Problems based on the framework by Li and Zhang (2009)

$$\begin{cases} f_1(x, t) = \alpha_{A,1}(x_I, t) + \beta_{A,1}(x_{II} - g_A(x_I, t), t) \\ \vdots \\ f_M(x, t) = \alpha_{A,M}(x_I, t) + \beta_{A,M}(x_{II} - g_A(x_I, t), t) \end{cases}$$

- $\alpha_{A,i}(x_I, t)$: PF-associated function for objective i
 - $\beta_{A,i}(x, t)$: PS-associated function for objective i
 - $g_A(x_I, t)$: distance-related function to the PF
- Some characteristics generated by changing three functions



Performance Measures

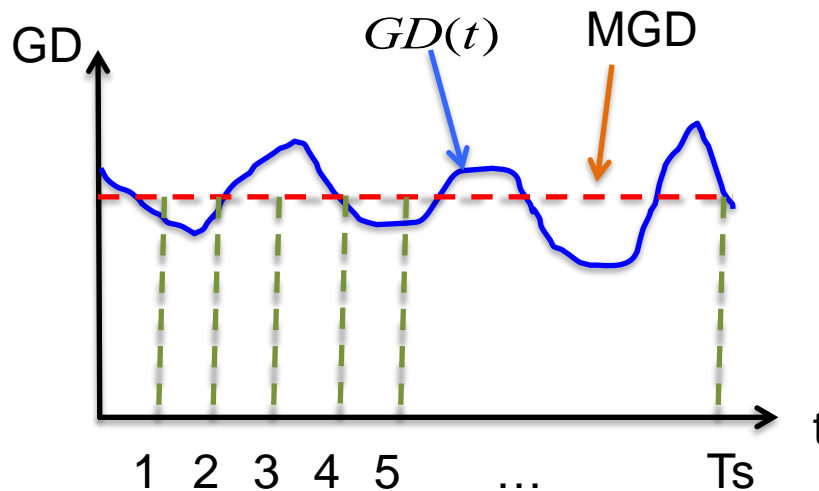
- For static MOPs, performance measures focus on
 - Convergence: Generational distance (GD), Inverted GD (IGD), C-metric,
 - Diversity: Spacing, maximum spread,
 - Comprehensive: Hypervolume (HV),
- For DMOPs, more measured aspects and indicators
 - Averaged measure values of a sequence of static period
 - Mean GD, Mean IGD, Mean SP, Mean HV,
 - Behavior-based performance measures
 - Reactivity
 - Stability
 - Robustness
 - ...

Performance Measures: Examples

- Mean of generational distance (MGD)

$$MGD = \frac{1}{T_s} \sum_{t=1}^{T_s} GD(t)$$

- T_s : number of time steps
- $GD(t)$: generational distance value at time t



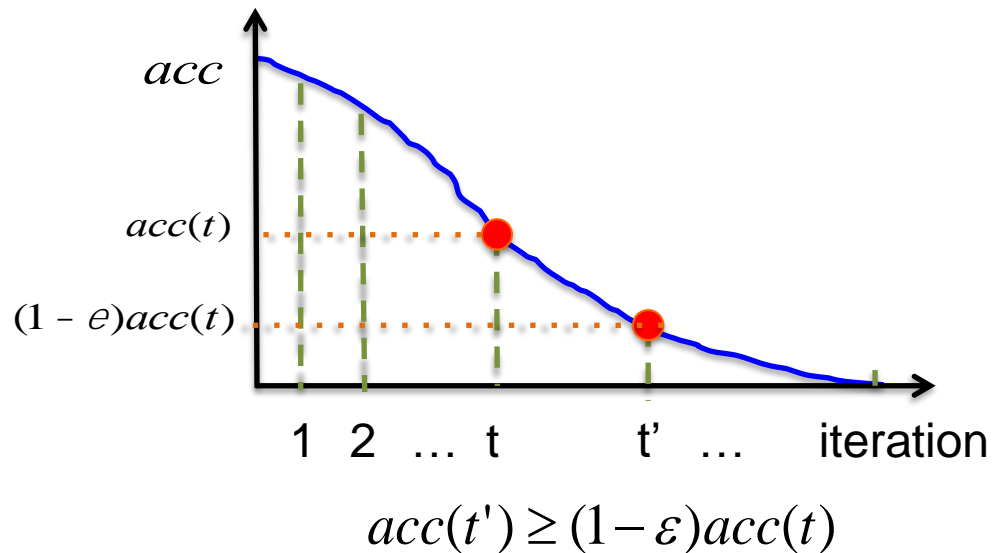
- Similarly, mean value of other measures can be defined

Performance Measures: Examples

- **Reactivity**: how long it takes to reach a specified accuracy threshold (ε):

$$react(t, \varepsilon) = \min \{t' - t \mid t < t' < t_{\max}, acc(t') \geq (1 - \varepsilon)acc(t)\}$$

- t_{\max} : the maximum number of iterations/generations



Performance Measures: Examples

- **Accuracy**: How well an approximation Pareto front (PF*) represents the true PF
- Accuracy often accounts for both diversity and convergence
- Hypervolume (HV) is preferred in definition of accuracy, which measures the HV difference between PF* and PF at time t:

$$acc(t) = |HV(PF^*) - HV(PF)|$$

- It can also be defined as the ratio of HV(PF*) to HV(PF):

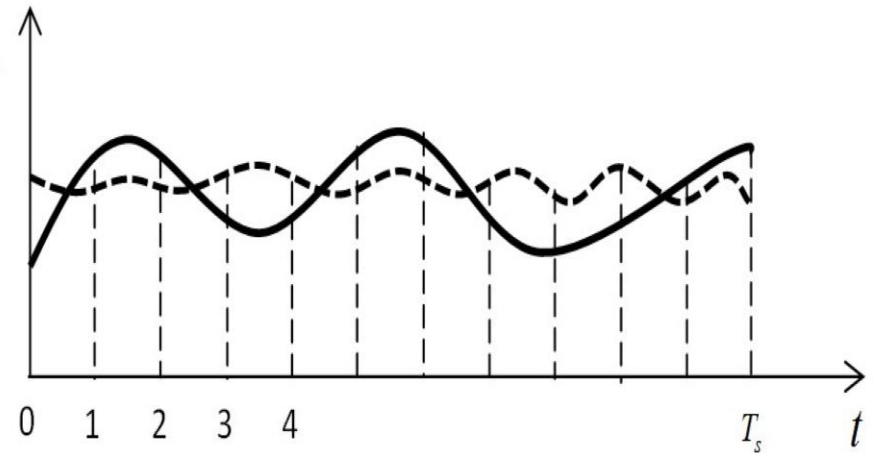
$$acc(t) = \frac{HV(PF^*)}{HV(PF)}$$

Performance Measures: Examples

- **Robustness**: used to describe the stability of the performance of an algorithm in a number of environmental changes, defined as follows:

$$R(PM) = \sqrt{\frac{1}{T_s - 1} \sum_{t=1}^{T_s} (PM_t - \overline{PM})^2}$$

where PM_t is the value of a performance metric at time t .



Part II: Approaches, Issues & Future Work

- Enhanced EC approaches for DMOPs
- Case studies
- Challenging issues and future work

EC for DMOPs: Things to Do

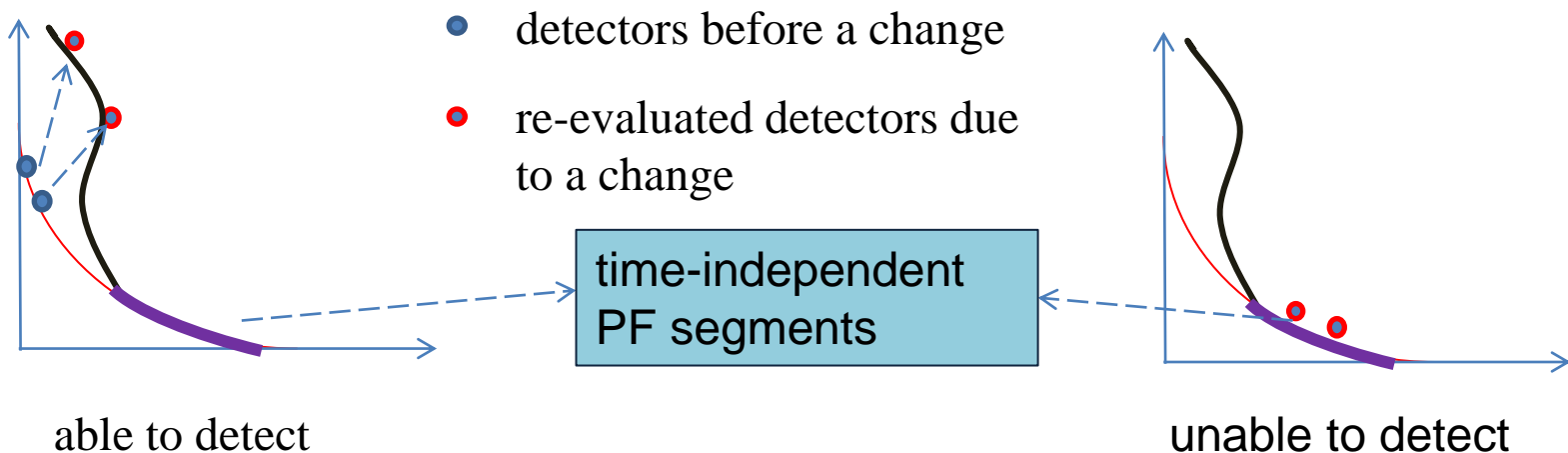
- To detect potential environmental changes
 - Success rate of detection
 - Cost of detection
- To track the changing PS/PF
 - To expect a steady and fast change response
 - To reduce the cost of tracking (given the budget limit, i.e., time, memory)

Change Detection Approaches

- Why is detection important ?
 - When a change occurs, archived solutions may become outdated
 - EAs would get misled if archived solutions are not re-evaluated in time
- Two ways of detecting changes:
 - **Individual-level detection**: fast but not robust
 - **Population-level detection**: slow but robust
 - Both methods could fail to detect changes (**not 100% guaranteed**)

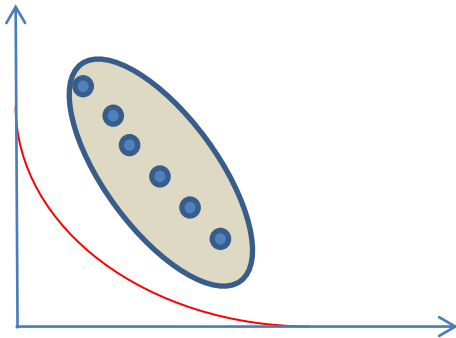
Individual-level Change Detection

- Re-evaluate some individuals' objective values every generation
 - Check the discrepancy between their current and previous fitness
- Success rate of detection depends on
 - Detectability of environmental changes
 - Location of detectors placed

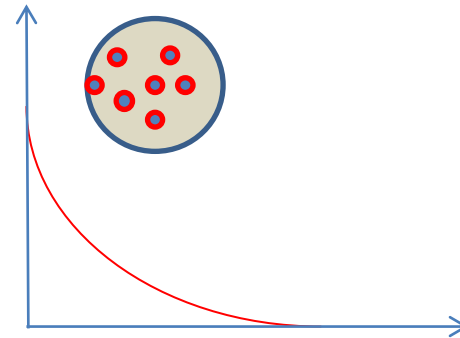


Population-level Change Detection

- Population-related statistical information, i.e., distribution, is assessed in every generation
 - Check the significance of variation in statistical information



population distribution
before a change



population distribution
after a change

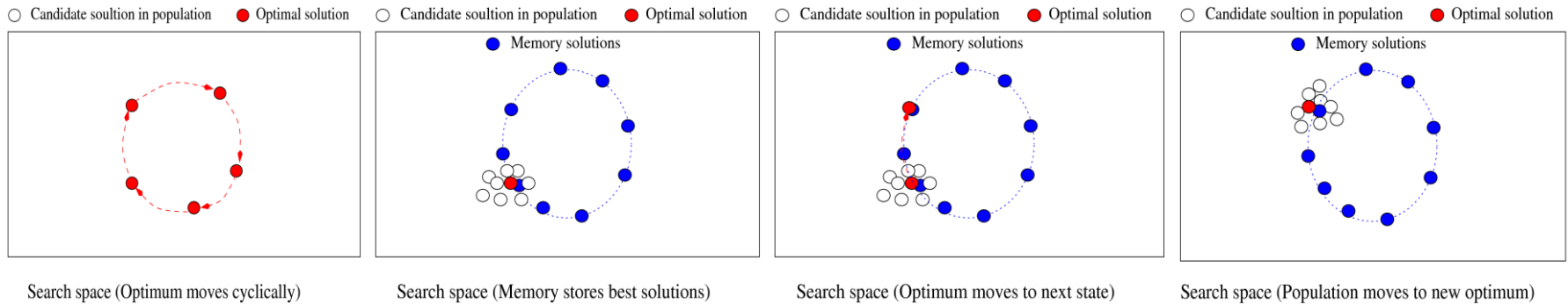
- Less sensitive to noise but possibly higher computational cost

Response Approaches

- How about restarting an EA after a change?
 - Natural and easy choice
 - But, not good choice due to:
 - It may be inefficient, wasting computational resources
 - It may lead to very different solutions before and after a change. For real-world problems, we may expect solutions to remain similar
- Extra approaches are needed to enhance EAs for DMOPs
- Typical approaches:
 - Memory: store and reuse useful information
 - Diversity: handle convergence directly
 - Multi-population: co-operate sub-populations
 - Prediction: predict where and when a change will occur

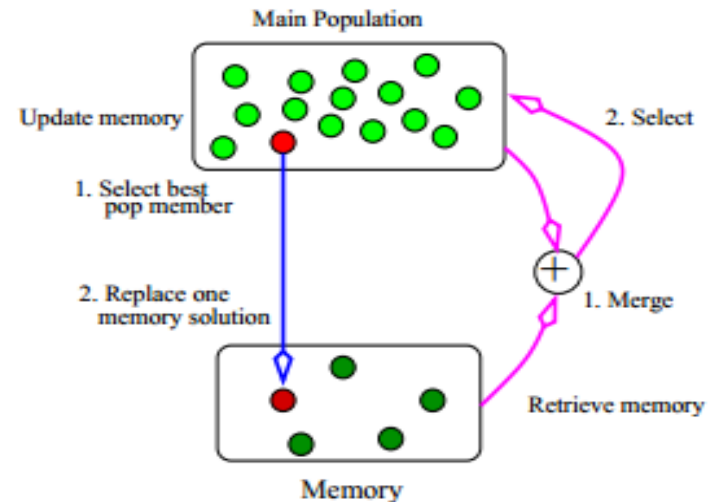
Memory-based Approaches

- Cyclic DMOPs: optimal solutions repeatedly return
- Memory: store history information for future use



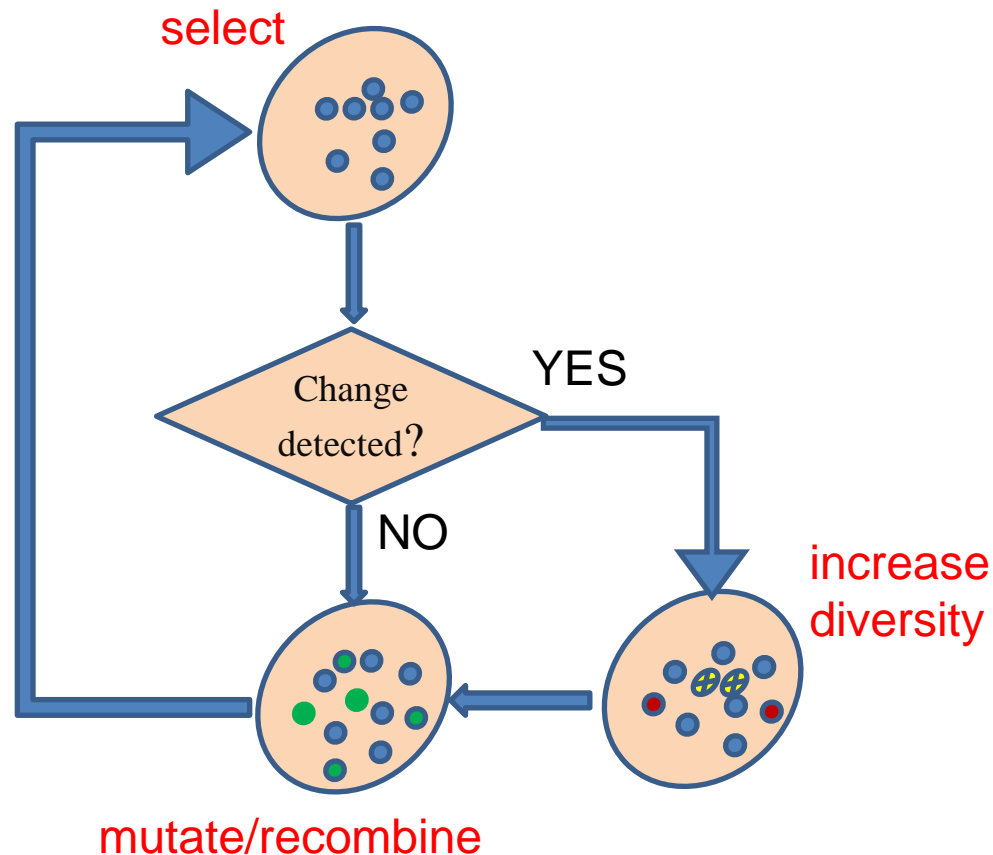
- Key issues:

- What information to store?
- When and how to retrieve memory?
- How to update memory?



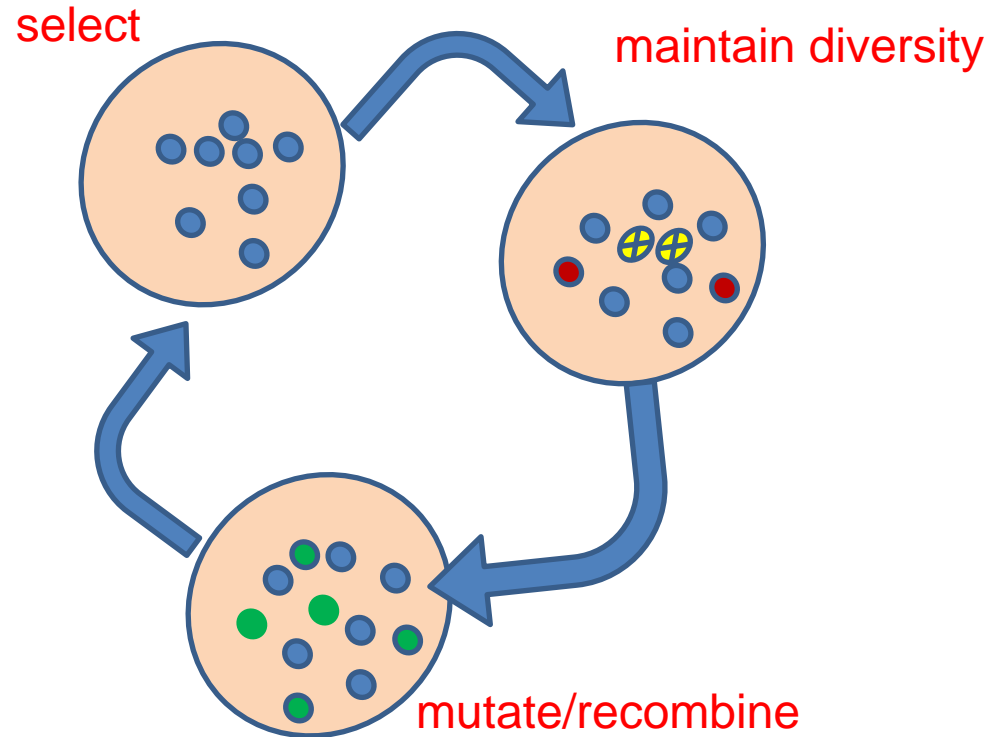
Diversity-based Approaches

- **Diversity increase:** introduce diversity after a change
 - Partially random restart, hyper-mutation, variable local search



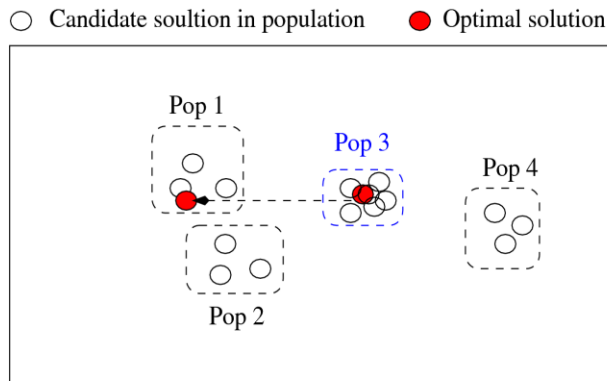
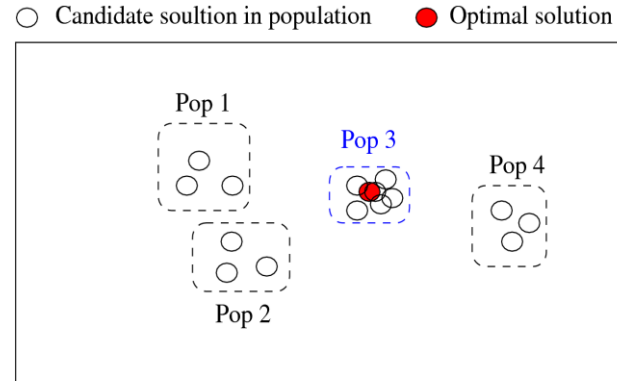
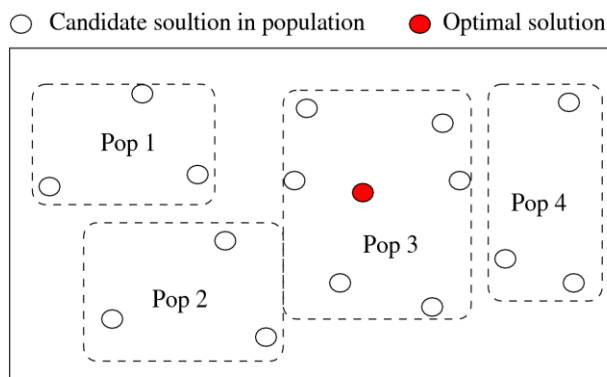
Diversity-based Approaches

- **Diversity maintenance:** maintain diversity throughout the run (even if no change occurs)
 - Random immigrants

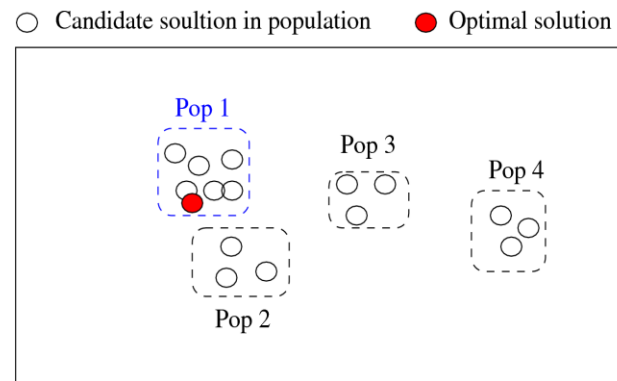


Multi-population Approaches

- Idea: Use several cooperative populations
 - Populations evolve independently in different areas of search space
 - Populations exclude each other to avoid overlap
 - When optimum moves, nearby population will take action



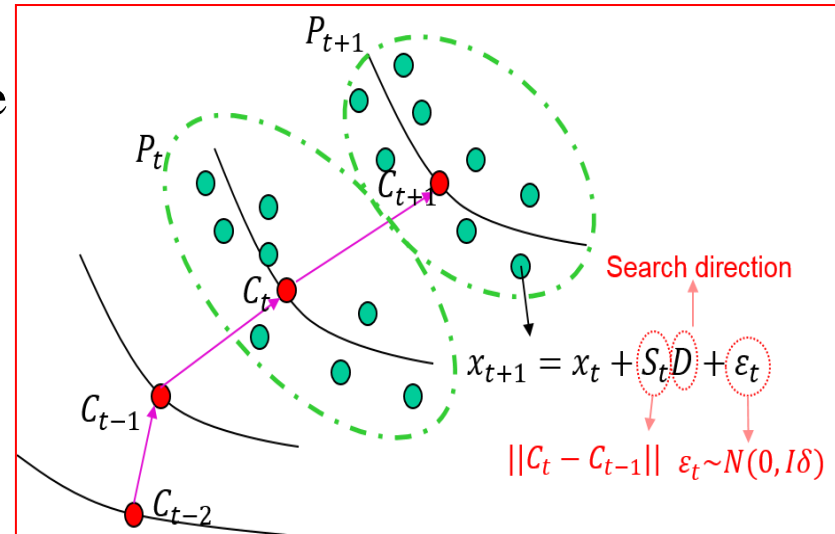
Search space (Optimum moves to near Pop 1)



Search space (Pop 1 becomes major pop)

Prediction Approaches

- For some DMOPs, changes exhibit predictable patterns
- Often to predict:
 - The location of new PS after a change
 - When the next change may occur
 - How much a change will be



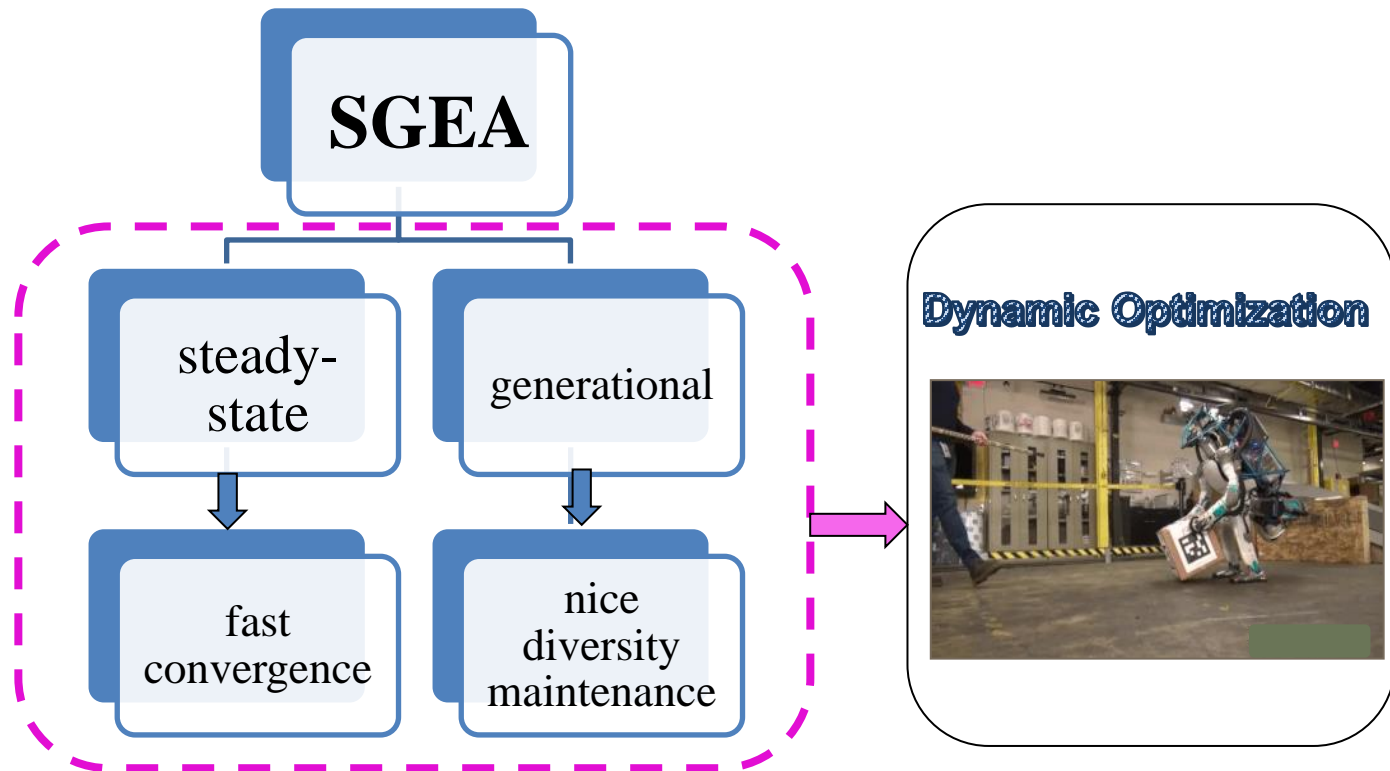
- Techniques:
 - Kalman filter (Muruganantham et al. 2016)
 - Population prediction strategy (Zhou et al. 2014)
 - Feed-forward prediction (Hatzakis & Wallace 2006)
 - Directed search strategy (Wu et al. 2015)
 - Evolutionary gradient search (Koo et al. 2010)
 - Center and knee points prediction (Zou et al. 2017)

Remarks on Enhancing Approaches

- No clear winner among the approaches
- Memory is efficient for cyclic environments
- Multi-population is good for multimodal problems
 - Able to maintain diversity
 - The search ability will decrease if too many sub-populations
- Diversity schemes are usually useful
 - Guided immigrants may be more efficient
- **Thumb of rule:** balancing **exploration** and **exploitation** over time

Case Study: EA for Continuous DMOPs

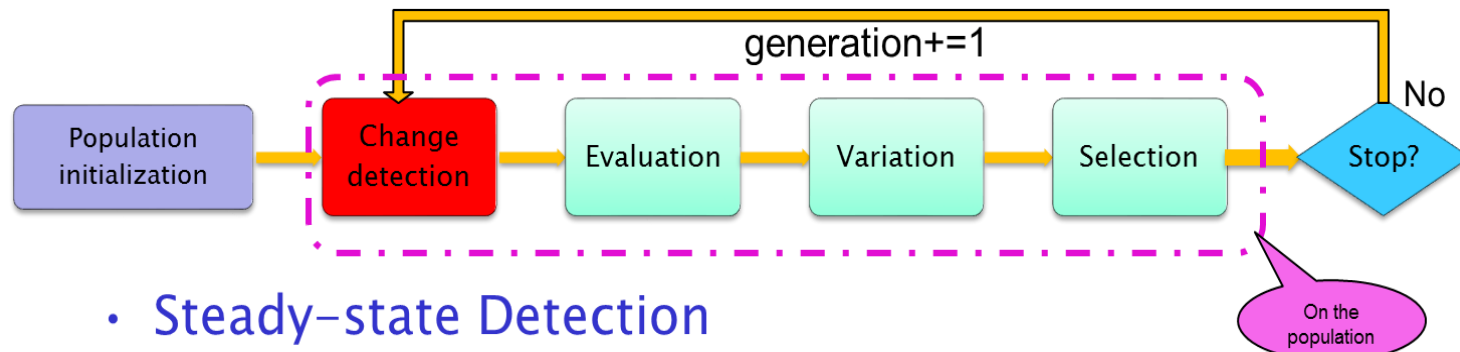
- Steady-Generational EA (SGEA)
 - Proposed by Jiang & Yang (2017b)
 - Hybrid of steady-state and generational methods
 - Novel steady-state change detection



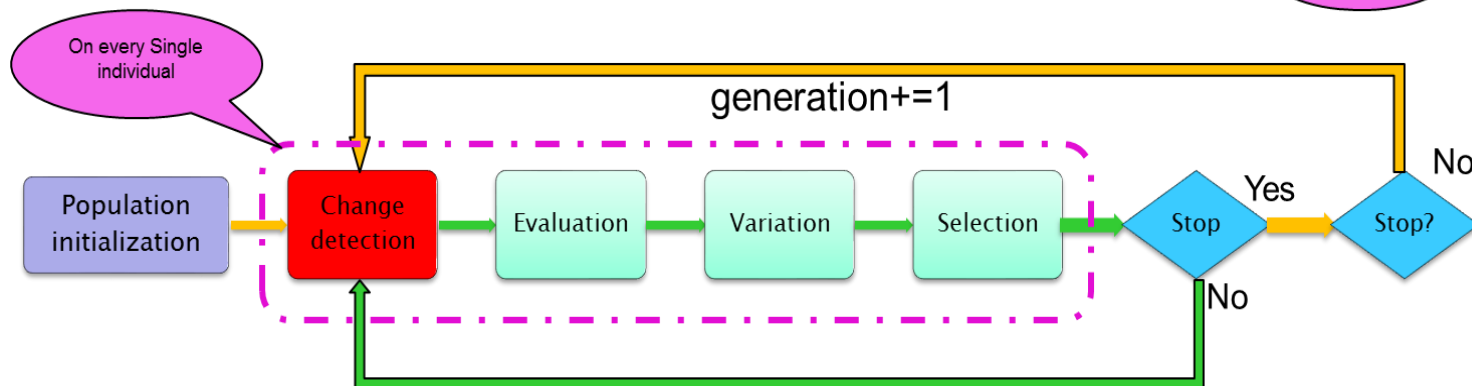
Case Study: EA for Continuous DMOPs

- Steady-state detection in SGEA
 - Can detect changes in the middle of generation
 - Can detect a change immediately
 - Rendering a fast follow-up action

- **Generational Detection**

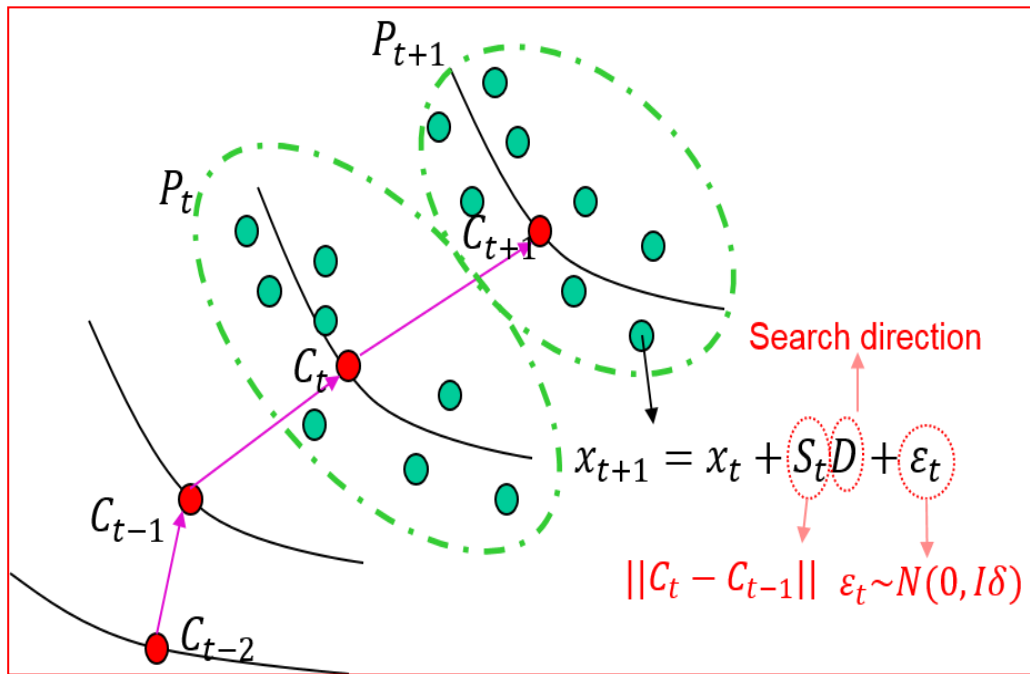


- **Steady-state Detection**

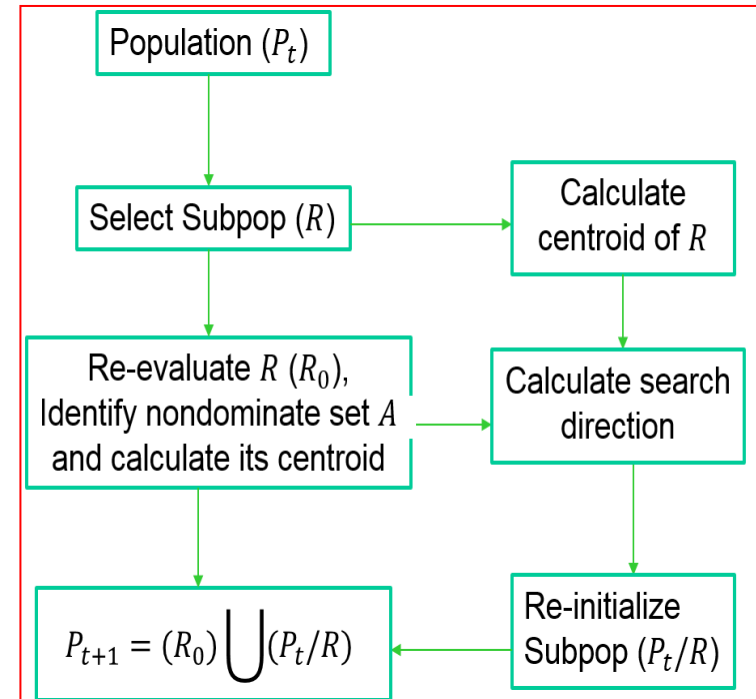


Case Study: EA for Continuous DMOPs

- Change response in SGEA:
 - Split pop into two subpops
 - Re-evaluate subpop1 (R) and keep its solutions
 - Re-initialize subpop2 by prediction methods



Movement of population



Procedure of change reaction

Case Study: EA for Continuous DMOPs

- Framework of SGEA

Algorithm 1 Framework of SGEA

```
1: Input:  $N$  (population size)
2: Output: a series of approximated POFs
3: Create an initial parent population  $P := \{x_1, \dots, x_N\}$ ;
4:  $(A, \bar{P}) := \text{EnvironmentSelection}(P)$ ;
5: while stopping criterion not met do
6:   for  $i := 1$  to  $N$  do
7:     if change detected and not responded then
8:        $\text{ChangeResponse}()$ ;
9:     end if
10:     $y := \text{GenerateOffspring}(P, A)$ ;
11:     $(P, A) := \text{UpdatePopulation}(y)$ ;
12:  end for
13:   $(A, \bar{P}) := \text{EnvironmentSelection}(P \cup \bar{P})$ ;
14:  Set  $P := \bar{P}$ ;
15: end while
```

steady-state

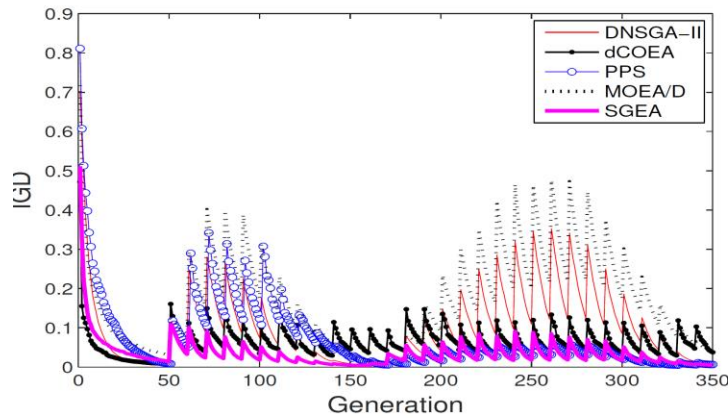
generational

Case Study: EA for Continuous DMOPs

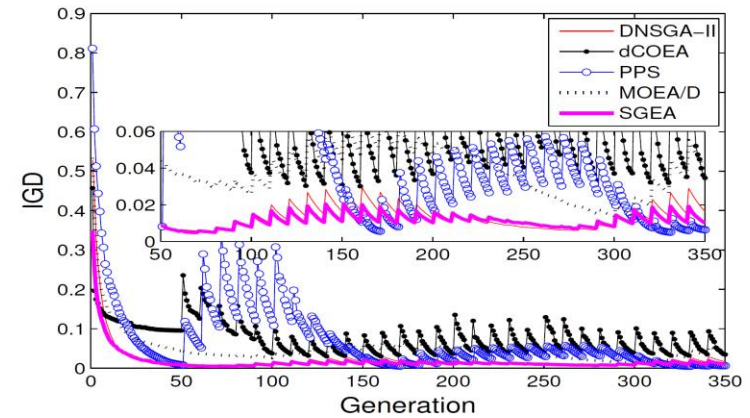
- Empirical study of SGEA
 - Test problems: FDA, dMOP, UDF, ...
 - Frequency of change: every 5, 10, 20 generations
 - Compared algorithms:
 - DNSGA-II: dynamic NSGAI (Deb et al. 2007)
 - dCOEA: Multi-population approach (Goh & Tan 2009)
 - PPS: population prediction strategy (Zhou et al. 2014)
 - MOEA/D: decomposition-based method (Zhang & Li 2007)
- Main findings:
 - Better tracking results in less frequently changing environments
 - SGEA shows high performance and outperforms other algorithms
 - But, SGEA fails in severe diversity loss due to changes
 - Introducing some random solutions can avoid diversity loss

Case Study: EA for Continuous DMOPs

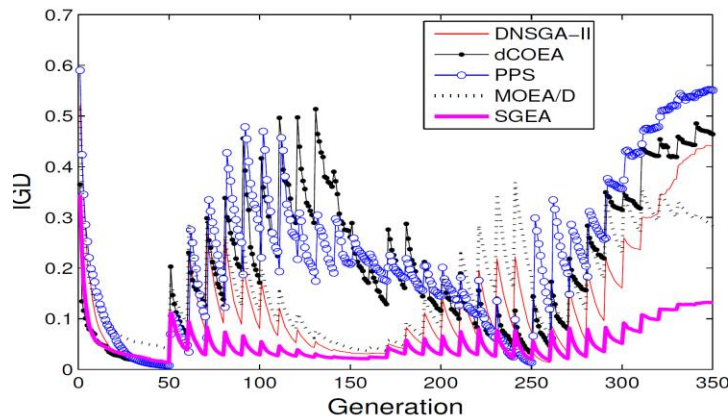
- Some results for FDA problems
 - Performance measure: IGD
 - SGEA is robust



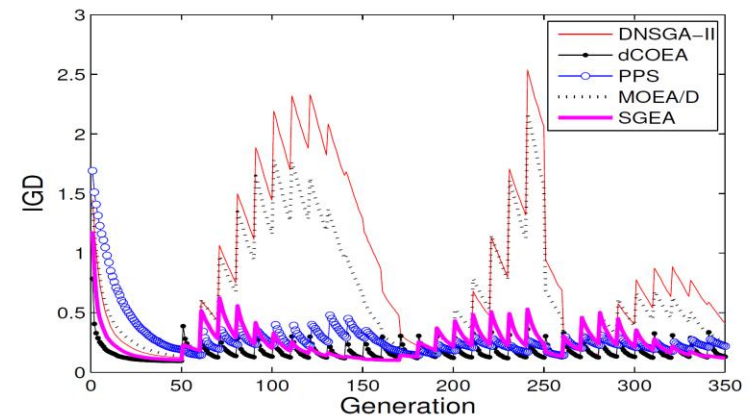
FDA1



FDA2



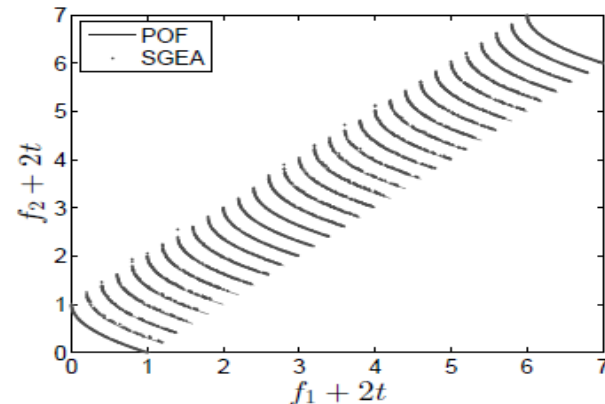
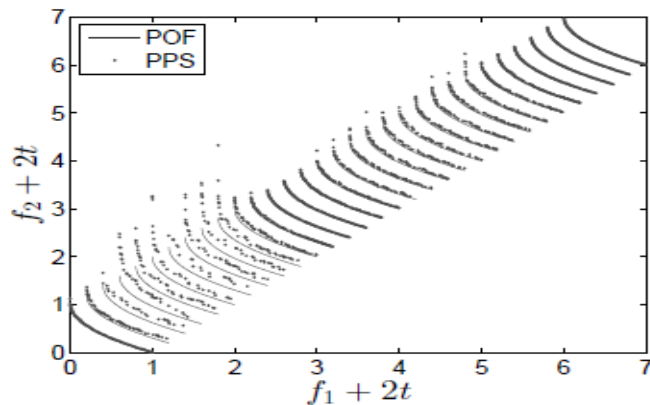
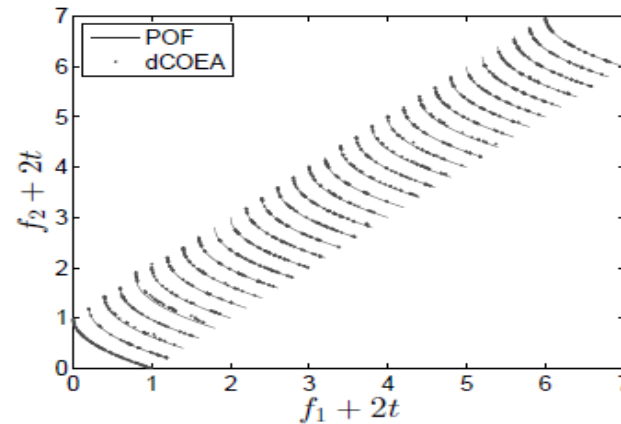
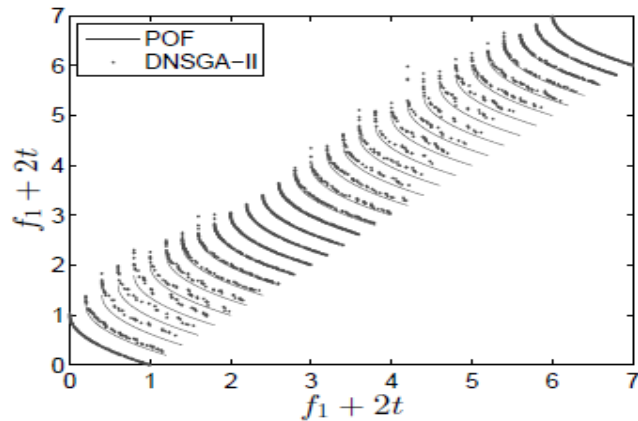
FDA3



FDA4

Case Study: EA for Continuous DMOPs

- Some results for FDA1
 - PF approximations obtained by algorithms
 - SGEA is able to track every change

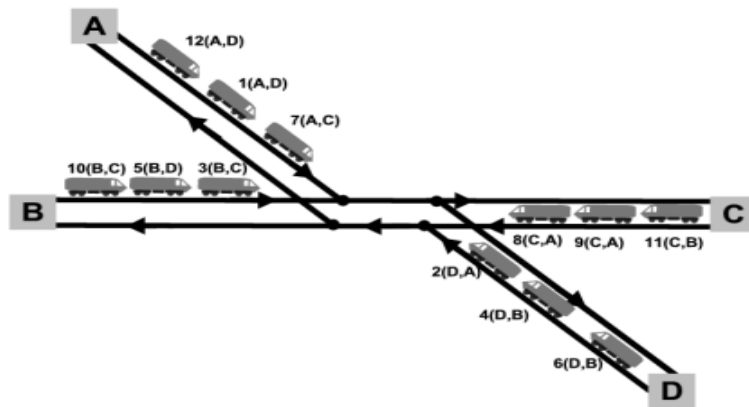


Case Study: ACO for DMOPs

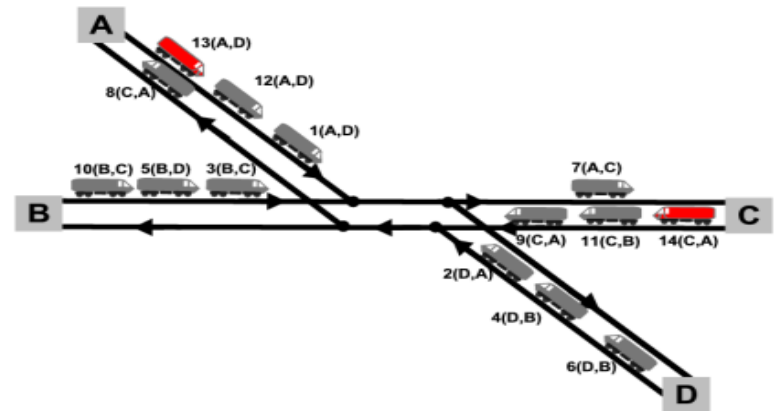
- ACO mimics the behaviour of ants searching for food
- ACO was first proposed for travelling salesman problems (TSPs) (Dorigo *et al.*, 1996)
- Generally, ACO is suitable for graph optimization problems, such as TSPs and vehicle routing problems (VRPs)
- The idea: let ants “walk” on the arcs of graph while “reading” and “writing” pheromones until they converge into a path
- Standard ACO consists of two phases:
 - Forward mode: Construct solutions
 - Backward mode: Pheromone update
- **Conventional ACO cannot adapt well to DMOPs due to stagnation behaviour**

Case Study: ACO for DM-RJRP

- Dynamic multi-objective railway junction re-scheduling problem (**DM-RJRP**):
 - To find a sequence of trains to pass through two junctions (North Stafford and Stenson) on the Derby to Birmingham line under delays
 - Two objectives:
 - Minimising timetable deviation, minimising additional energy expenditure
 - Dynamics:
 - As trains are waiting to be rescheduled at the junction, more timetabled trains will be arriving, which will change the nature of the problem



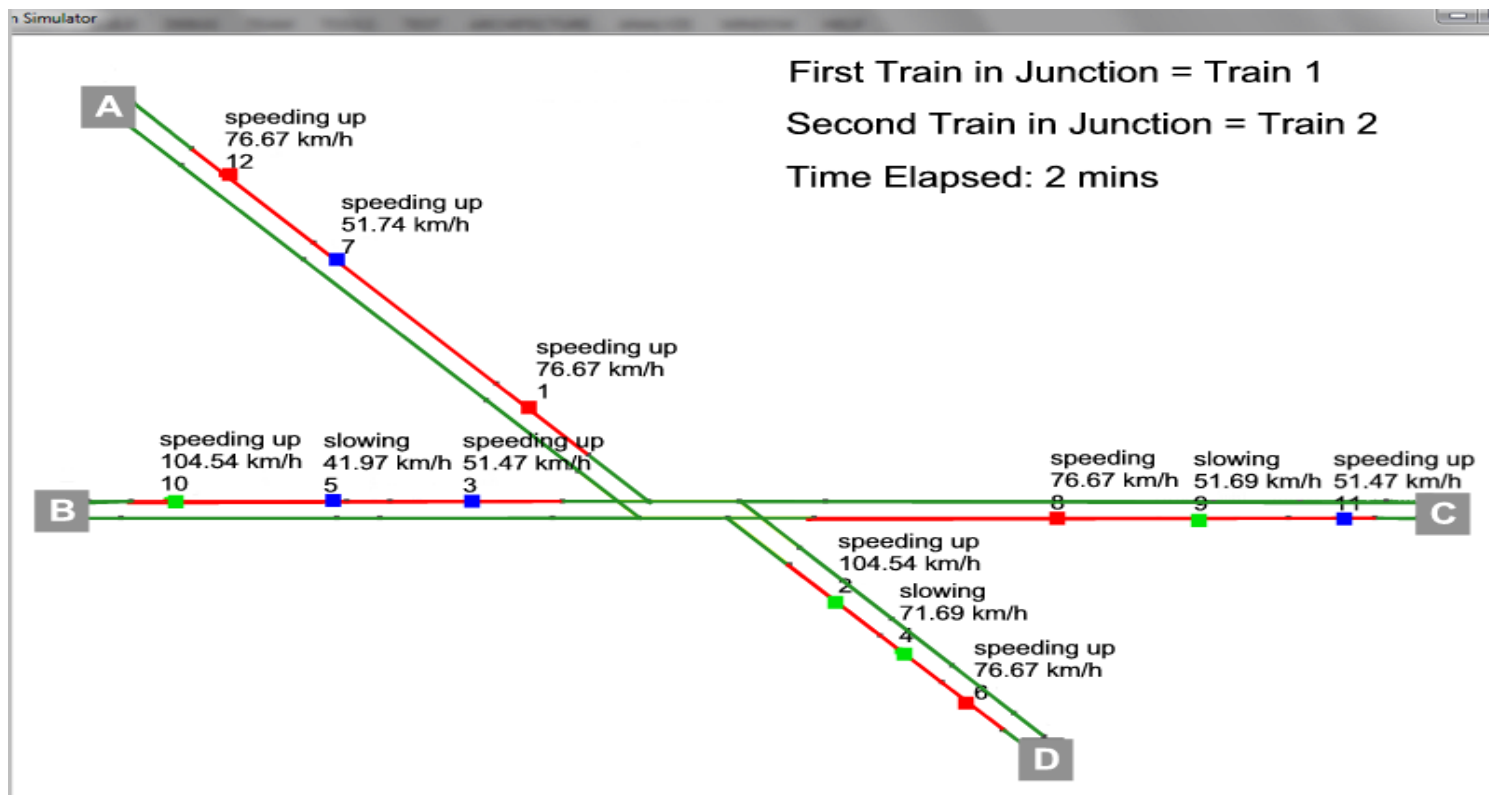
Junction before a change



Junction after a change

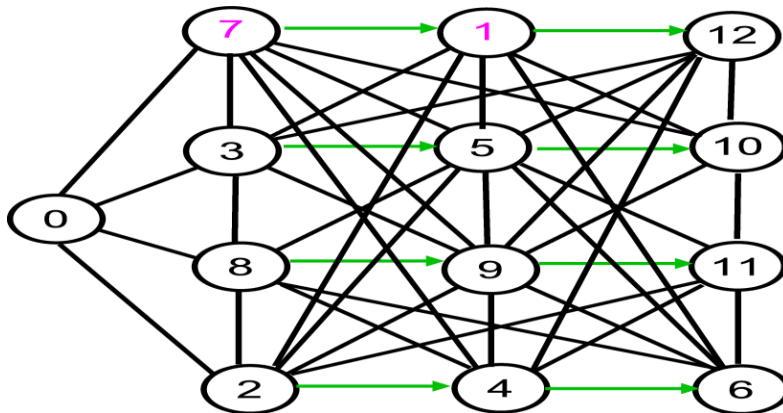
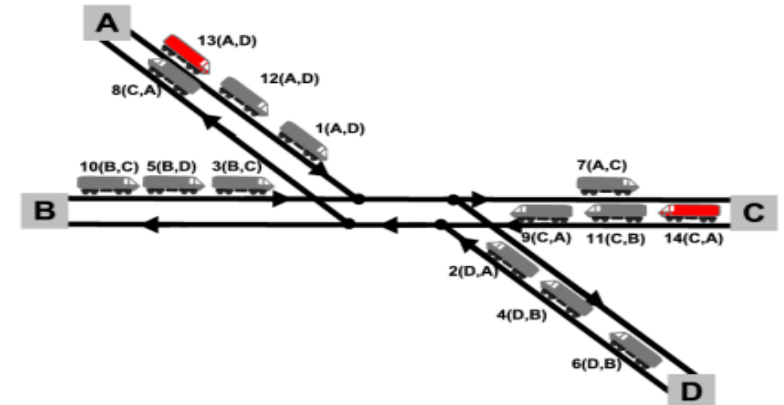
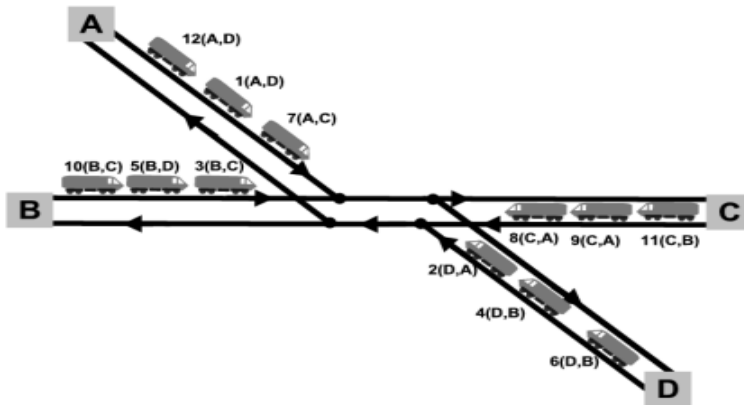
Case Study: ACO for DM-RJRP

- The North Stafford and Stenson junctions train simulator:
 - Developed using C++ Visual Studio 2012
 - Dynamism:
 - Introduced to the simulator by adding m trains at a time interval f (minutes), where m represents the magnitude of change and f the frequency of change

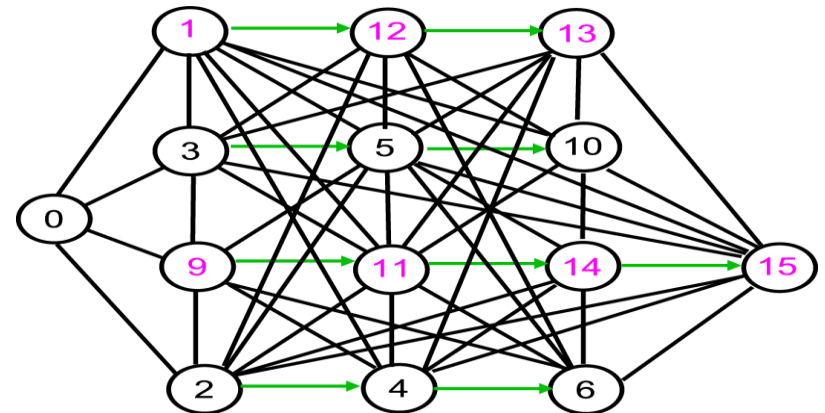


Case Study: ACO for DM-RJRP

- ACO for DM-RJRP: a graphical representation
 - A fully connected, partially one-directional, weighted graph
 - Each node represents a train
- All ants are initially placed at an imaginary start node (zero)



Node matrix before a change



Node matrix after a change

Case Study: Proposed ACO for DM-RJRP

- **DM-PACO**: a new version of population-based ACO (P-ACO)
 - A pheromone and heuristic matrix for each objective
 - An archive to store non-dominated solutions (repaired after a change)
 - A memory: created from the archive and re-created after a change
- **DM-MMAS**: a new version of Max-Min Ant System (MMAS)
 - A pheromone matrix for each objective
 - An archive to store non-dominated solutions
 - Four designs based on clearing archive or pheromones after a change

FOUR DIFFERENT VERSIONS OF THE DM-MMAS ALGORITHM

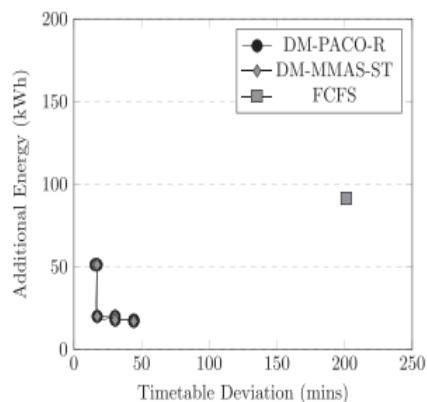
	Clear Pheromones	Retain Pheromones
Clear Archive	DM-MMAS-SC	DM-MMAS-ST
Retain Archive	DM-MMAS-NC	DM-MMAS-NT

- Peer algorithms: NSGA-II and FCFS (First Come First Served)

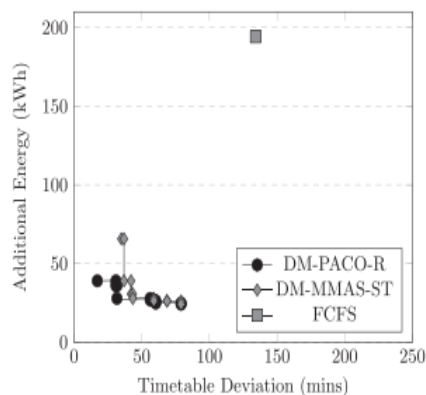
Case Study: Proposed ACO for DM-RJRP

● Findings:

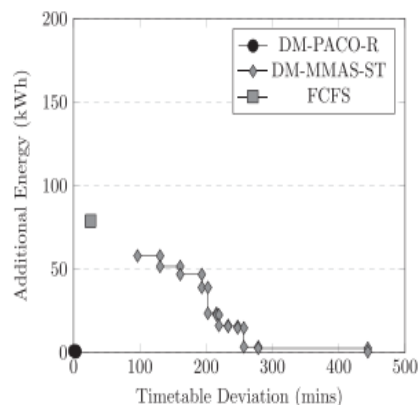
- All ACO algorithms can find a POS of solutions for the DM-RJRP
- DM-PACO outperformed DM-MMAS algorithms
- DM-PACO also outperformed NSGA-II and FCFS
- For large and frequent changes:
 - Good to retain an archive of non-dominated solutions
 - Good to update pheromones for new environments
- Interaction between objectives are more complex than expected



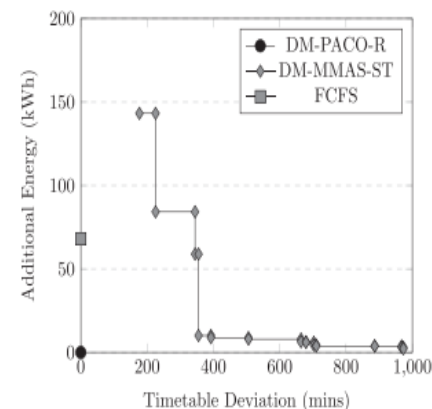
(a) Change 0



(b) Change 1



(c) Change 4



(d) Change 6

EC for DMOPs: Challenging Issues

- Detecting changes:
 - Most studies assume that changes are easy to detect or visible to an algorithm whenever occurred
 - In fact, changes are difficult to detect for many DMOPs
- Understanding the characteristics of DMOPs:
 - What characteristics make DMOPs easy or difficult?
 - Little work, needs much more effort
- Analysing the behaviour of EC methods for DMOPs:
 - Requiring more theoretical analysis tools
 - Big question: **Which EC methods for what DMOPs?**
- Real world applications:
 - How to model real-world DMOPs?
 - How to extend the applicability of EC methods?

Future Work

- The domain has attracted a growing interest recently
 - But, far from well-studied
- New approaches needed: esp. hybrid approaches
- Theoretical analysis: greatly needed
- Real world applications: also greatly needed
 - Fields: logistics, transport, MANETs, data streams, social networks, ...



Summary

- EC for DMOPs: challenging but important
- The domain is still young and active:
 - Benchmarking and performance measures
 - Optimization approaches
 - Theoretic study
 - Real-world applications
- More young researchers are greatly welcome!



Thanks!

Relevant Information

- IEEE CIS Task Force on EC in Dynamic and Uncertain Environments
 - <http://ieee-tf-ecidue.cug.edu.cn/>
- Personal publications and source codes:
 - <http://www.tech.dmu.ac.uk/~syang/publications.html>
- Two EPSRC funded projects on EC for DOPs
 - “EAs for DOPs: Design, Analysis and Applications”
 - Funding/Duration: over £600K/3.5 years (1/2008–7/2011)
 - <http://gtr.rcuk.ac.uk/project/B807434B-E9CA-41C7-B3AF-567C38589BAC>
 - “EC for Dynamic Optimisation in Network Environments”
 - Funding/Duration: ~£1M/4.5 years (2/2013–8/2017)
 - <http://gtr.rcuk.ac.uk/project/C43F34D3-16F1-430B-9E1F-483BBADCD8FA>

References-1

- S. Biswas, S. Das, P.N. Suganthan, C.A. Coello Coello (2014). Evolutionary multiobjective optimization in dynamic environments: A set of novel benchmark functions, IEEE CEC 2014, pp. 3192-3199.
- Q. Chen, J. Ding, S. Yang, T. Chai (2020). A novel evolutionary algorithm for dynamic constrained multiobjective optimization problems. IEEE Trans Evol Comput, in press.
- K. Deb, N. U. B. Rao, S. Karthik (2007). Dynamic multi-objective optimization and decision-making using modified NSGA-II: A case study on hydro-thermal power scheduling. in Evolutionary Multi-Criterion Optimization, pp. 803-817.
- J. Eaton, S. Yang, M. Gongora (2017). Ant colony optimization for simulated dynamic multi-objective railway junction rescheduling. IEEE Trans Intell Transport Syst, 18(11): 2980-2992.
- M. Farina, K. Deb, P. Amato (2004). Dynamic multiobjective optimization problems: test cases, approximations, and applications. IEEE Trans Evol Comput, 8(5): 425–442.
- S. B. Gee, K. C. Tan, H. A. Abbass (2017). A Benchmark Test Suite for Dynamic Evolutionary Multiobjective Optimization. IEEE Trans on Cybern, 47(2): 461-472.
- C. Goh, K. C. Tan (2009). A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. IEEE Trans Evol Comput, 13(1): 103–127.
- D. Gong, B. Xu, Y. Zhang, Y. Guo, S. Yang (2020). A similarity-based cooperative co-evolutionary algorithm for dynamic interval multi-objective optimization problems. IEEE Trans Evol Comput, 24(1): 142-156.

References-2

- I. Hatzakis, D. Wallace (2006). Dynamic multi-objective optimization with evolutionary algorithms: A forward-looking approach. GECCO 2006, pp. 1201-1208.
- M. Helbig, A. P. Engelbrecht (2014). Benchmarks for dynamic multi-objective optimisation algorithms. ACM Comput Surv 46(3): 37:1–37:39.
- Y. Hu, J. Zheng, J. Zou, S. Yang, J. Ou, R. Wang (2020). A dynamic multi-objective evolutionary algorithm based on intensity of environmental change. Inf. Sci, 523:49-62.
- S. Huband, P. Hingston, L. Barone, L. While (2006). A review of multiobjective test problems and a scalable test problem toolkit. IEEE Trans Evol Comput 10(5): 477-506.
- S. Jiang, M. Kaiser, S. Yang, S. Kollias, and N. Krasnogor (2019). A scalable test suite for dynamic multiobjective optimisation. IEEE Trans Cybern, in press.
- S. Jiang, S. Yang (2017a). Evolutionary dynamic multi-objective optimization: Benchmarks and algorithm comparisons. IEEE Trans Cybern, 47(1): 198-211.
- S. Jiang, S. Yang (2017b). A steady-state and generational evolutionary algorithm for dynamic multi-objective optimization. IEEE Trans Evol Comput, 21(1): 65-82.
- S. Jiang, M. Kaiser, S. Yang, S. Kollias, N. Krasnogor (2020). A scalable test suite for dynamic multiobjective optimisation. IEEE Trans Cybern, 50(6): 2814-2826.
- Y. Jin, B. Sendhoff (2004). Constructing dynamic optimization test problems using the multi-objective optimization concept. in Applications of Evol. Comput., pp. 525-536.
- Y. Jin, J. Branke (2005). Evolutionary optimization in uncertain environments—A survey. IEEE Trans Evol Comput, 9(3): 303–317.

References-3

- Q. Li, J. Zou, and S. Yang, J. Zheng, G. Ruan (2019). A predictive strategy based on special points for evolutionary dynamic multi-objective optimization. *Soft Computing*, 23(11): 3723-3739.
- Z. Liang, T. Wu, X. Ma, Z. Zhu, S. Yang (2020). A dynamic multi-objective evolutionary algorithm based on decision variable classification. *IEEE Trans Cybern*, in press.
- Z. Liang, S. Zheng, Z. Zhu, S. Yang (2019). Hybrid of memory and prediction strategies for dynamic multiobjective optimization. *Inform Sci*, 485: 200-218.
- H. Li and Q. Zhang (2009). Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, *IEEE Trans Evol Comput* 13(2): 284–302.
- W. T. Koo, C. Goh, K. Tan (2010): A predictive gradient strategy for multi-objective evolutionary algorithms in a fast changing environment. *Memet Comput* 2(2): 87–110.
- J. Mehnen, G. Rudolph, T. Wagner (2006). Evolutionary optimization of dynamic multiobjective functions. Tech Report CI-204/06. Universit at Dortmund, Germany
- A. Muruganantham (2017) Dynamic Multiobjective Optimization Using Evolutionary Algorithms. PhD Thesis.
- M. Orouskhani (2017). Dynamic Multiobjective Optimization Using Hybrid Algorithm of Cat Swarm Optimization and Borda Count Method. PhD Thesis.
- G. Ruan, G. Yu, J. Zheng, J. Zou, S. Yang (2017). The effect of diversity maintenance on prediction in dynamic multi-objective optimization. *Appl Soft Comput*, 58: 631-647.
- Y. Wang, B. Li (2009). Investigation of memory-based multi-objective optimization evolutionary algorithm in dynamic environment. *IEEE CEC*, pp. 630-637.

References-4

- Y. Wu, Y. Jin, X. Liu (2015). A directed search strategy for evolutionary dynamic multiobjective optimization. *Soft Comput*, 19(11): 3221-3235.
- S. Yang, X. Yao (2013). *Evolutionary Computation for Dynamic Optimization Problems*. Springer.
- Q. Zhang, S. Yang, S. Jiang, R. Wang, X. Li (2020). Novel prediction strategies for dynamic multi-objective optimization. *IEEE Trans Evol Comput*, 24(2): 260-274.
- Q. Zhang, A. Zhou, S. Zhao, P. Suganthan, W. Liu, S. Tiwari (2008). Multiobjective optimization test instances for the CEC 2009 special session and competition. School Comput. Sci. Elec. Eng., Univ. Essex, U.K., Nanyang Technol. Univ., Singapore, Tech. Rep. CES-487, pp. 1–30.
- A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, E. Tsang (2007). Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization. *EMO-2007*, pp. 832-846.
- A. Zhou, Y. Jin, Q. Zhang (2014). A population prediction strategy for evolutionary dynamic multiobjective optimization. *IEEE Trans Cybern*, 44(1): 40-53.
- J. Zou, Q. Li, S. Yang, H. Bui, J. Zheng (2017). A prediction strategy based on center points and knee points for evolutionary dynamic multi-objective optimization. *Appl Soft Comput*, 61: 806-818.
- J. Zou, Q. Li, S. Yang, J. Zheng, Z. Peng, T. Pei (2019). A dynamic multiobjective evolutionary algorithm based on a dynamic evolutionary environment model. *Swarm and Evol Comput*, 44: 247-259.