Evolutionary Computation for Dynamic Multi-objective Optimization

Shengxiang Yang

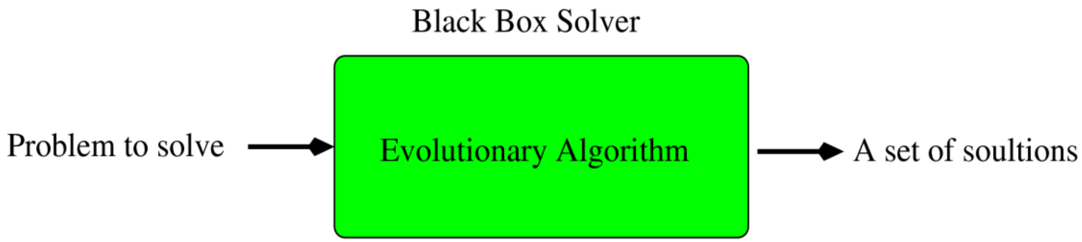
Centre for Computational Intelligence (CCI)
De Montfort University, Leicester LE1 9BH, UK
http://www.tech.dmu.ac.uk/~syang
Email: syang@dmu.ac.uk

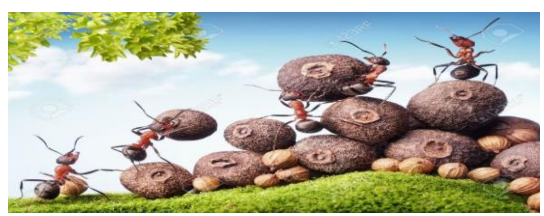
Outline of the Talk

- Concepts: Evolutionary computation (EC) and dynamic multi-objective optimization problems (DMOPs)
- Benchmark test problems and performance measures
- EC-based approaches for DMOPs
- Case studies and future directions
- Summary

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

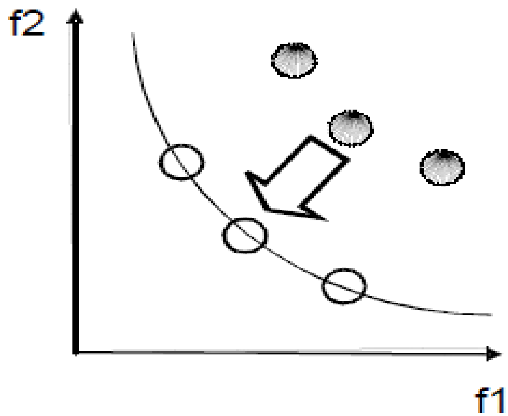
- Advantages of EC methods:
  - Multiple solutions in a single run
  - No strict requirements to problems
  - Easy to use

- Widely used for optimisation and search problems
  - Financial and economical systems
  - Transportation and logistics systems
  - Industry engineering
  - Automatic programming, art and music design
  - ......

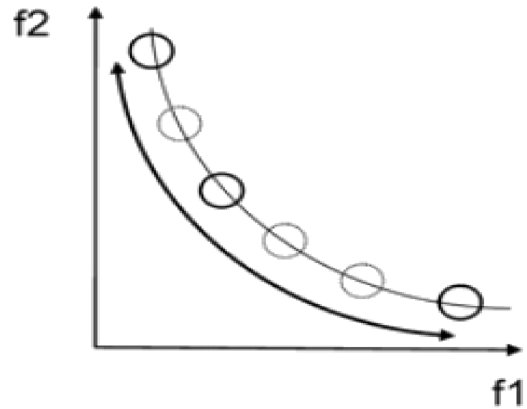
- EC has also been used for solving multi-objective optimisation problems

EC for MOPs

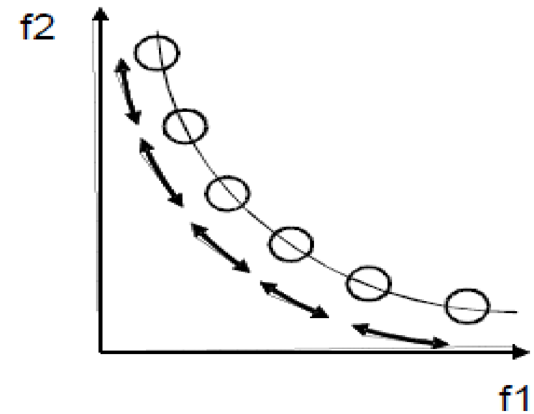
- Traditionally, research has focused on static MOPs:
  - Aim to find the POF with three requirements



Close to the Pareto front
(Convergence)



Widely spread
(Extensity)



Equally distributed
(Uniformity)

Diversity: combines extensity and uniformity

- Many EAs have been developed for MOPs over four decades
- 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:

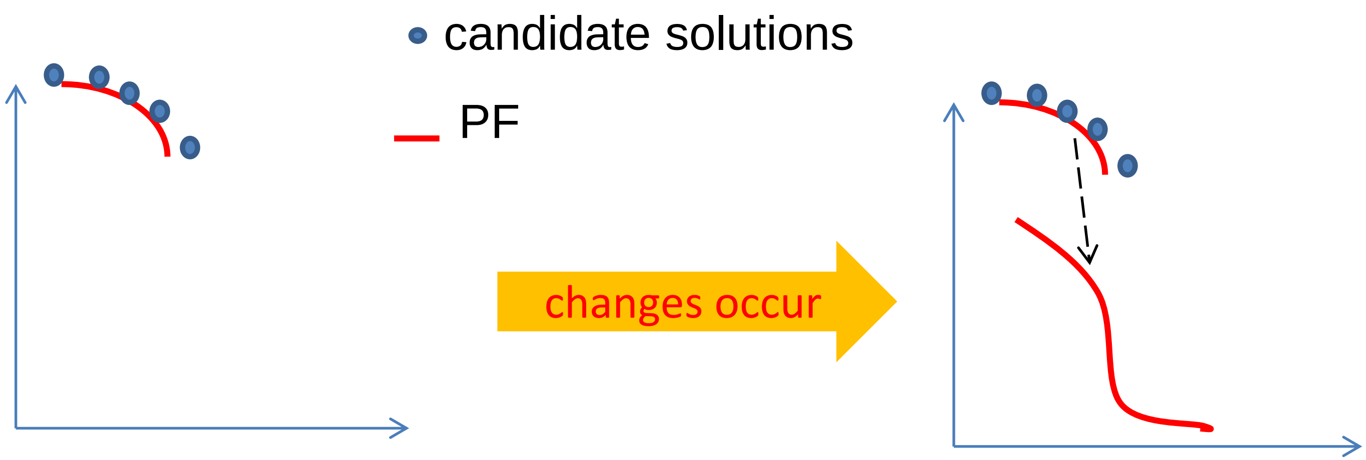
$$F=\left(f_1(x,\varphi,t),f_2(x,\varphi,t),\cdots,f_M(x,\varphi,t)\right)^T$$

  - x： decision variables;
  - j : parameter;
  - t ： time

- DMOPs: a special class of dynamic multiobjective problems that are solved by an algorithm as time precedes.

Why Are DMOPs Challenging?

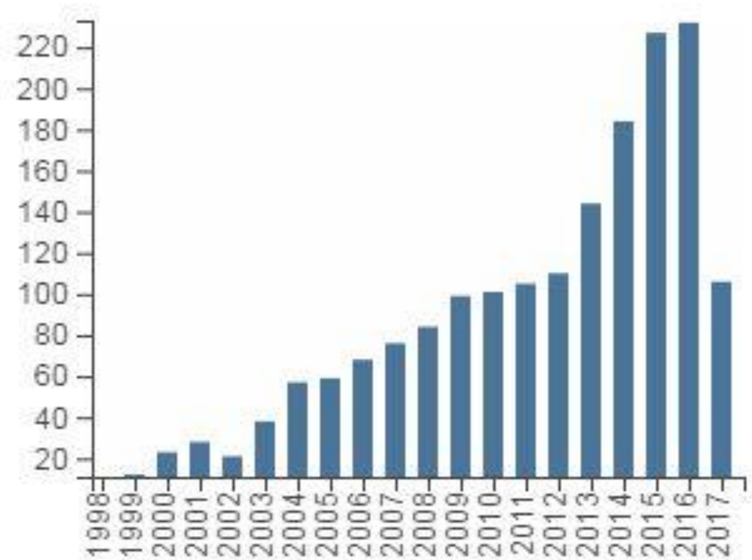
- For DMOPs, PFs and/or PSs may change over time
  - Challenge 1: need to track the moving PF/PS over time
  - Challenge 2: need to re-spread non-dominated solutions



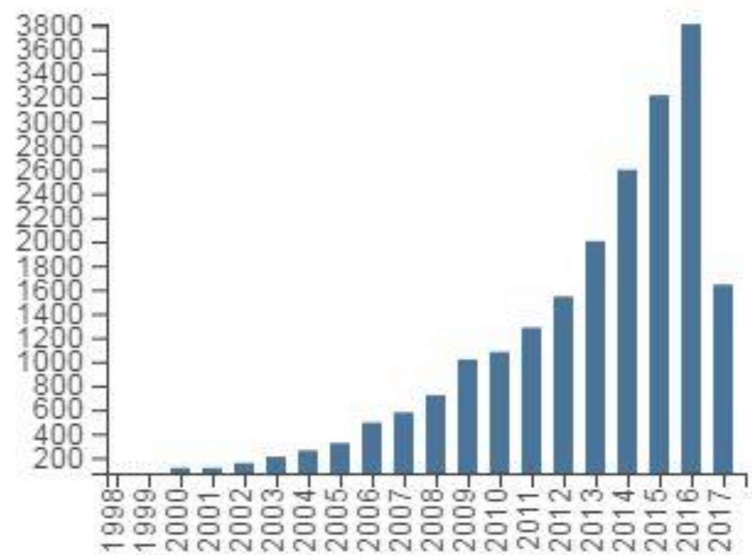
- DMOPs challenge traditional EAs
  - Limited time to respond to environmental changes
  - Once converged, hard to escape from an outdated PF/PS
  - Very likely to lose diversity after a change

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)



publication by year



citation 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
  - <u>Mixed Type (*Jiang & Yang 2017a*): All of the above four types of change can be present, either randomly or in turn</u>

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:

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

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

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

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

Continuous DMOP Benchmarks

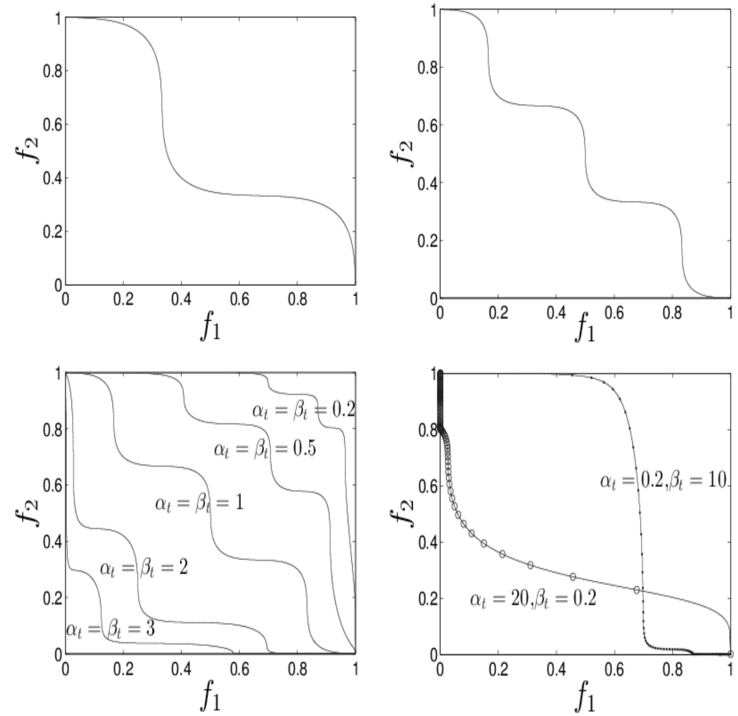
- A number of continuous DMOP benchmarks proposed:
  - Jin-Sendhoff's framework by Jin & Sendhoff (2004)
  - FDA test suite by Farina et al. (2004)
  - DSW test problems by Mehnen et al. (2006)
  - dMOP test suite by Goh & Tan (2007)
  - HE test suite by Helbig & Engelbrecht (2013, 2014)
  - UDF test 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)

JY Generator by Jiang & Yang (2017)

- Focusing on dynamics analysis

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

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



- 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

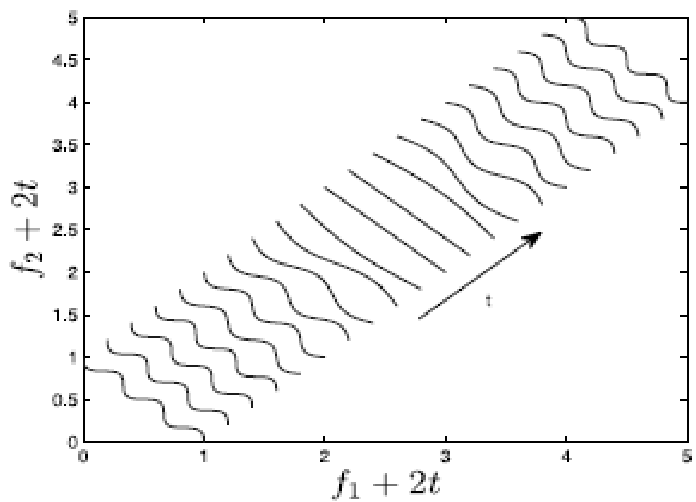
S. Jiang, S. Yang. Evolutionary dynamic multi-objective optimization: benchmarks and algorithm comparisons. IEEE Transactions on Cybernetics, 47(1): 198-211, 2017

JY Generator by Jiang & Yang (2017) - 2

- JY2: time-changing PS and PF

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

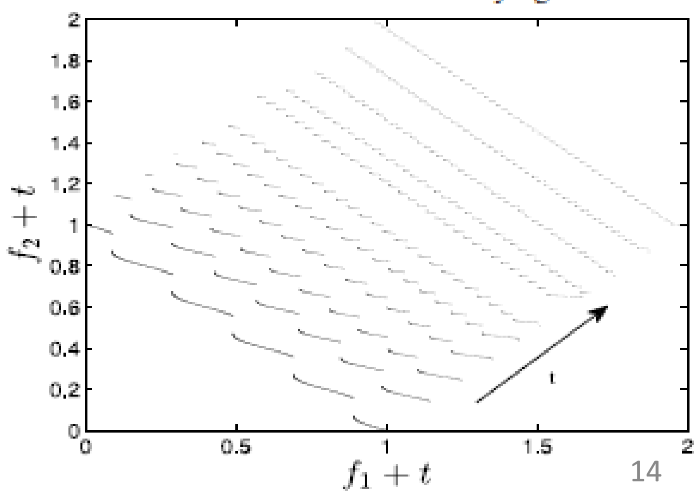
POF of $JY2$ with 21 time windows varying from 0 to 2.



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

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

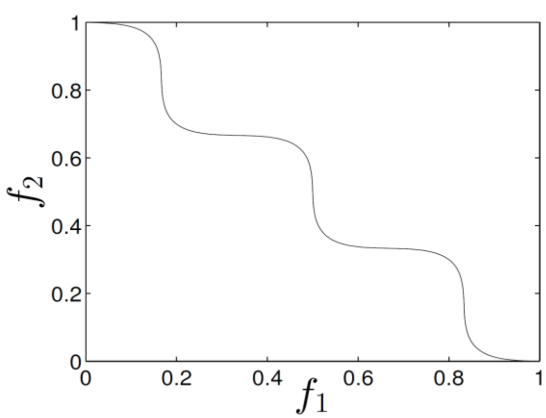
POF of $JY4$ with 11 time windows varying from 0 to 2.



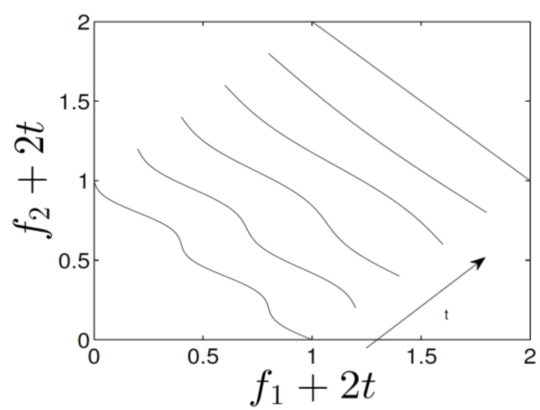
JY Generator by Jiang & Yang (2017) - 3

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

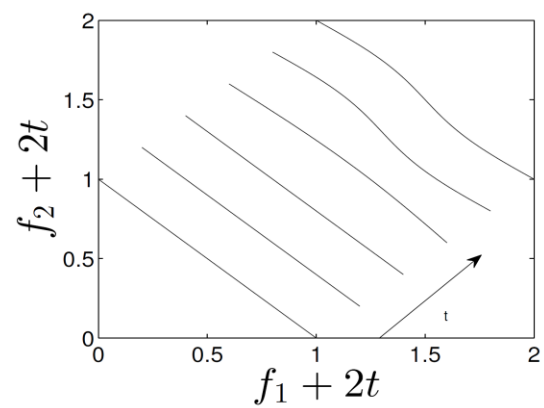
$$
JY10: \begin{cases}
min \quad F(\mathbf{x},t) = (f_1(\mathbf{x},t), f_2(\mathbf{x},t))^T \\
f_1(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(x_1 + A_t \sin(W_t \pi x_1))^{\alpha_t} \\
f_2(\mathbf{x},t) = (1+g(\mathbf{x_{II}},t))(1-x_1+A_t \sin(W_t \pi x_1))^{\beta_t} \\
g(\mathbf{x_{II}},t) = \sum_{x_i \in \mathbf{x_{II}}} (x_i+\sigma-G(t))^2, G(t) = |sin(0.5\pi t)| \\
A(t) = 0.05, \quad W(t) = 6 \\
\alpha_t = \beta_t = 1 + \sigma G(t), \sigma \equiv (\lfloor \frac{\tau}{\tau_t \rho_t} \rfloor + R) \pmod 3 \\
\mathbf{x_I} = (x_1) \in [0,1], \mathbf{x_{II}} = (x_2, \ldots, x_n) \in [-1,1]^{n-1},
\end{cases}
$$



Static PF, dynamic PS



Dynamic PF, dynamic PS



Dynamic PF, static PS

Performance Measures

- For static MOPs, performance measures focus on
  - Convergence: Generational distance (GD), Inverted GD (IGD), C-metric, ...
  - Diversity: Spacing, maximum spread, ...
  - Comprehensive: Hypervolume, ……

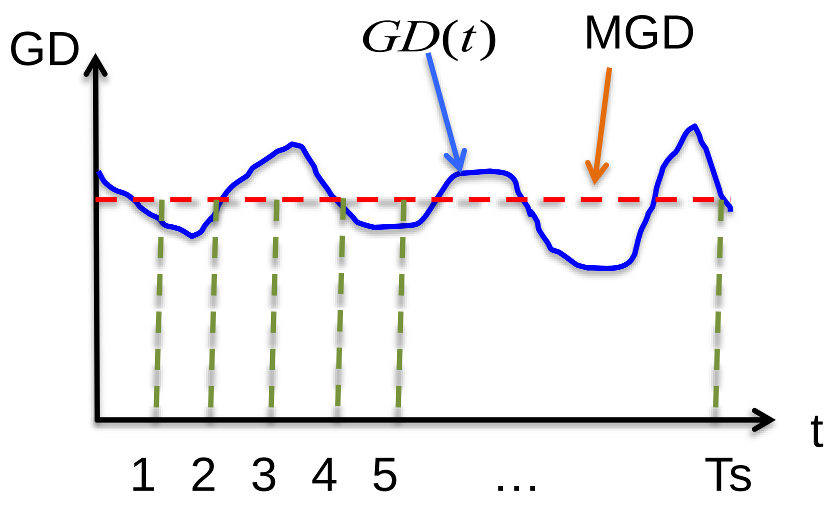
- For DMOPs, more measured aspects and indicators
  - Averaged measure values of a sequence of static period
    - Mean GD, Mean IGD, Mean SP, Mean HV, …....
  - Behavior-based performance measures
    - Reactivity
    - Stability
    - Robustness
    - ...

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

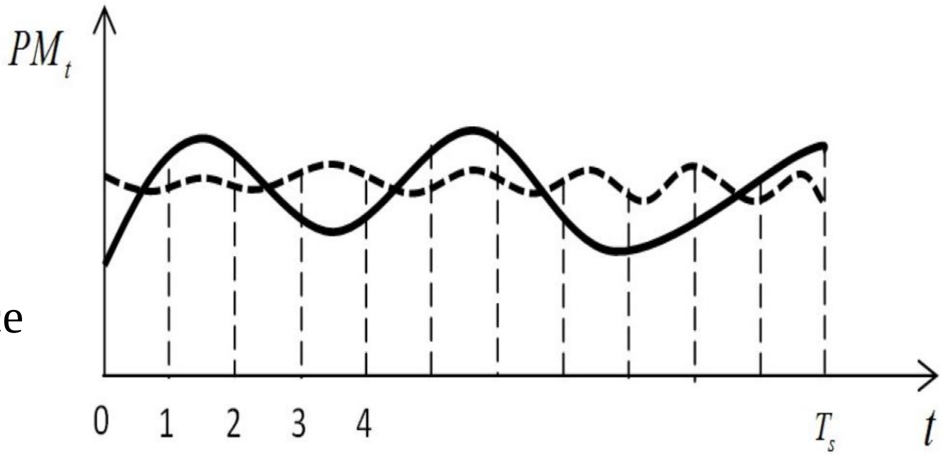


- Similarly, mean value of other measures can be defined

Performance Measures: Examples

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

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

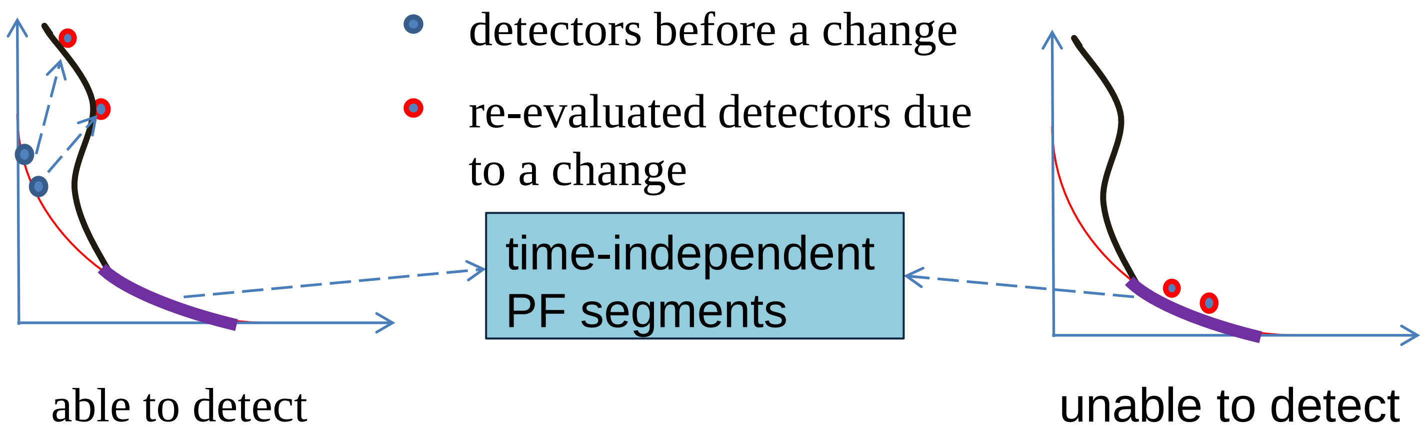


where PM_t is the value of a performance metric at time t.

S. Jiang, S. Yang. Evolutionary dynamic multi-objective optimization: benchmarks and algorithm comparisons. IEEE Transactions on Cybernetics, 47(1): 198-211, 2017

EC for DMOPs: Things to Do

- To detect potential environmental changes
  - Individual-level detection: fast but not robust
  - Population-level detection: slow but robust
  - Both methods could fail to detect changes (not 100% guaranteed)



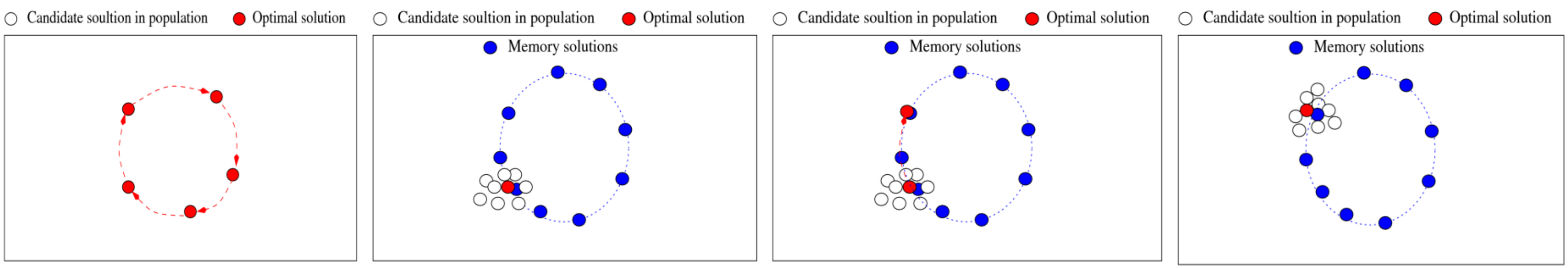
- To track the changing PS/PF
  - To expect a steady and fast change response
  - To reduce the cost of tracking (given a budget limit: time & memory)

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

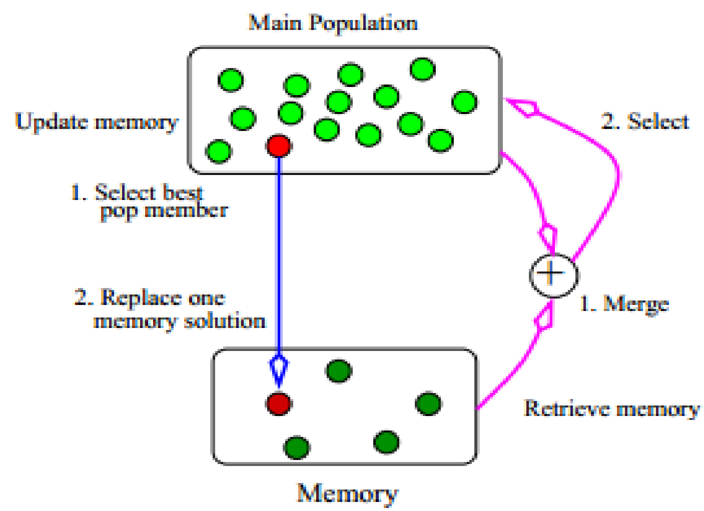
Memory-based Approaches

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



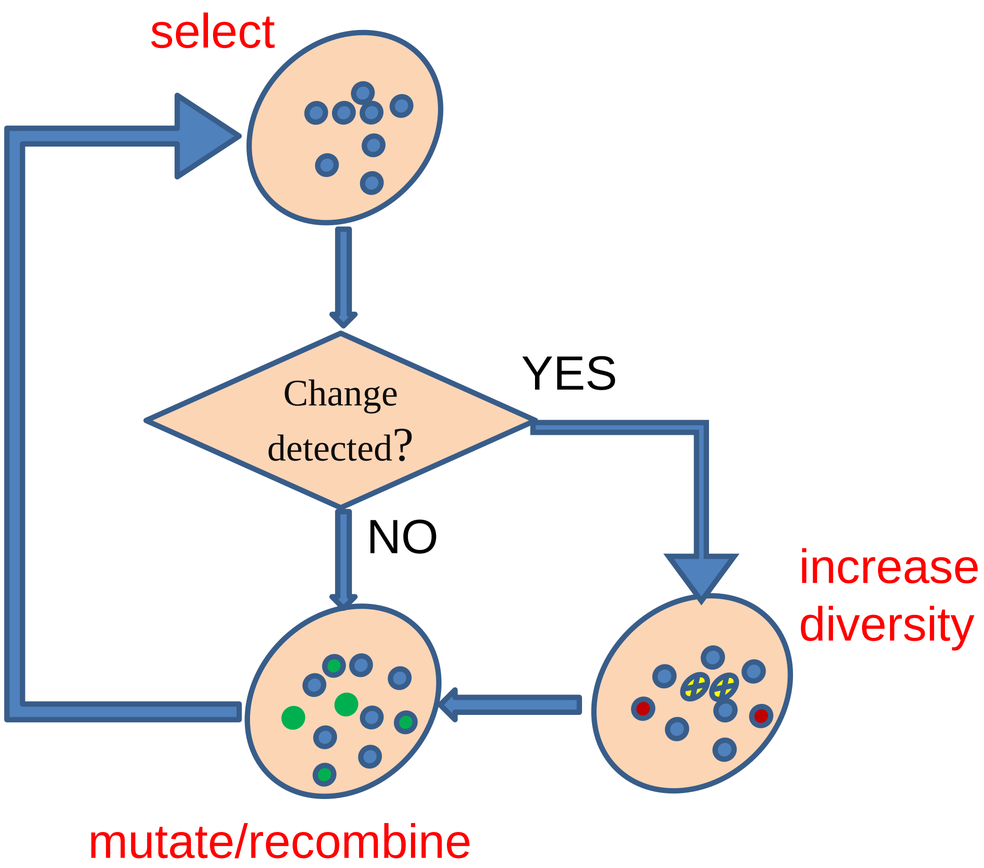
Search space (Optimum moves cyclically) Search space (Memory stores best solutions) Search space (Optimum moves to next state) Search space (Population moves to new optimum)

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



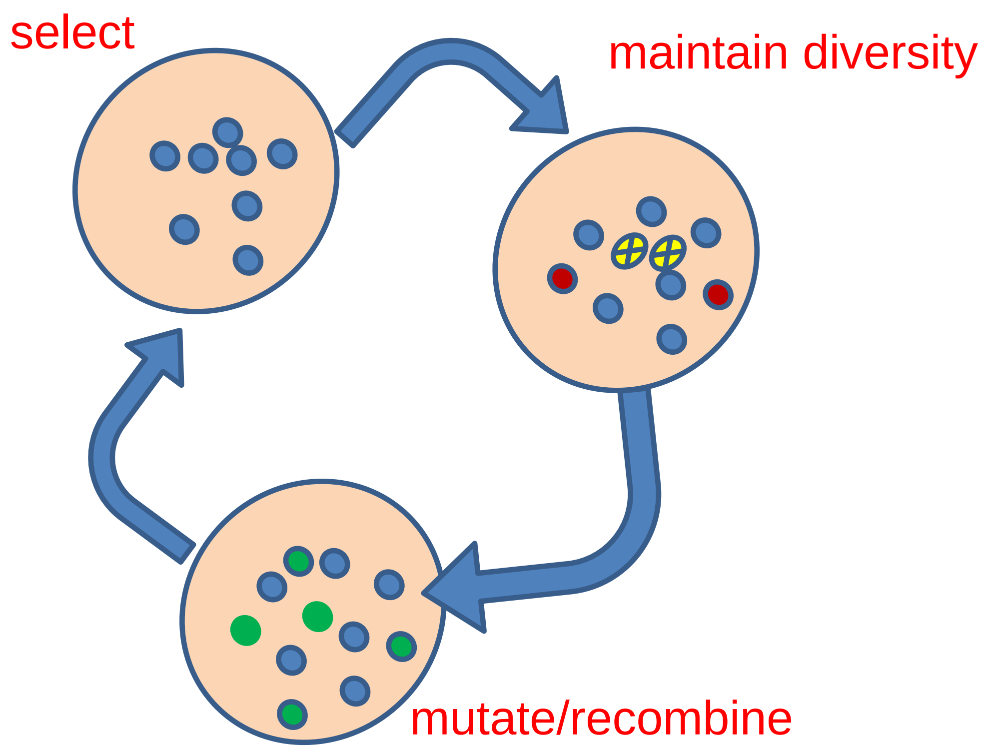
Diversity-based Approaches

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



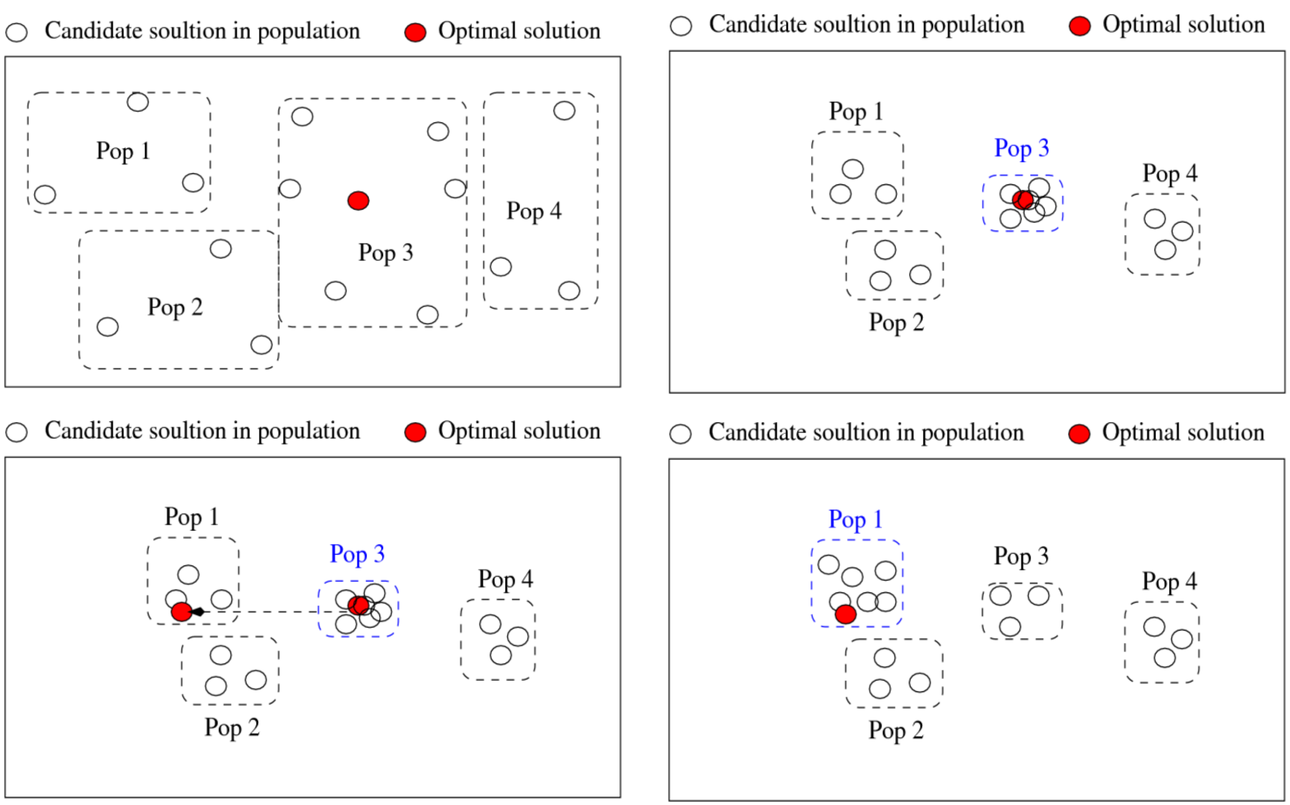
Diversity-based Approaches

- 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

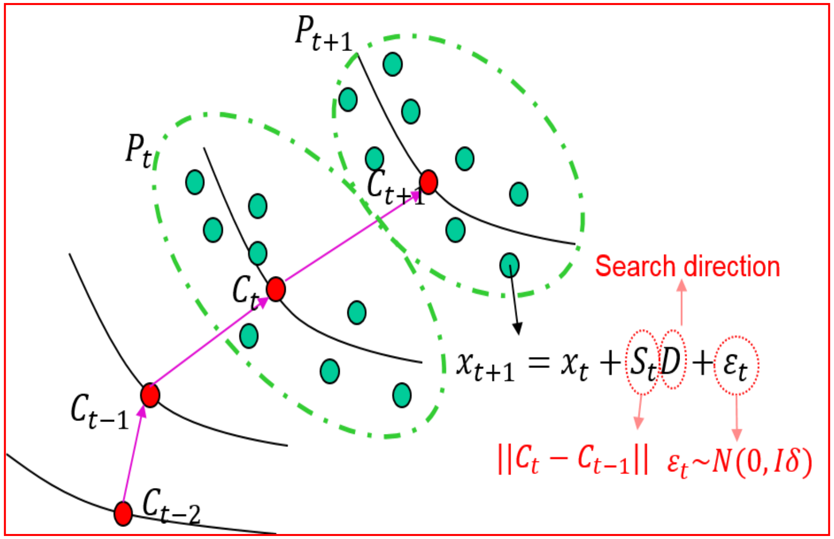


Search space (Optimum moves to near Pop 1)

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



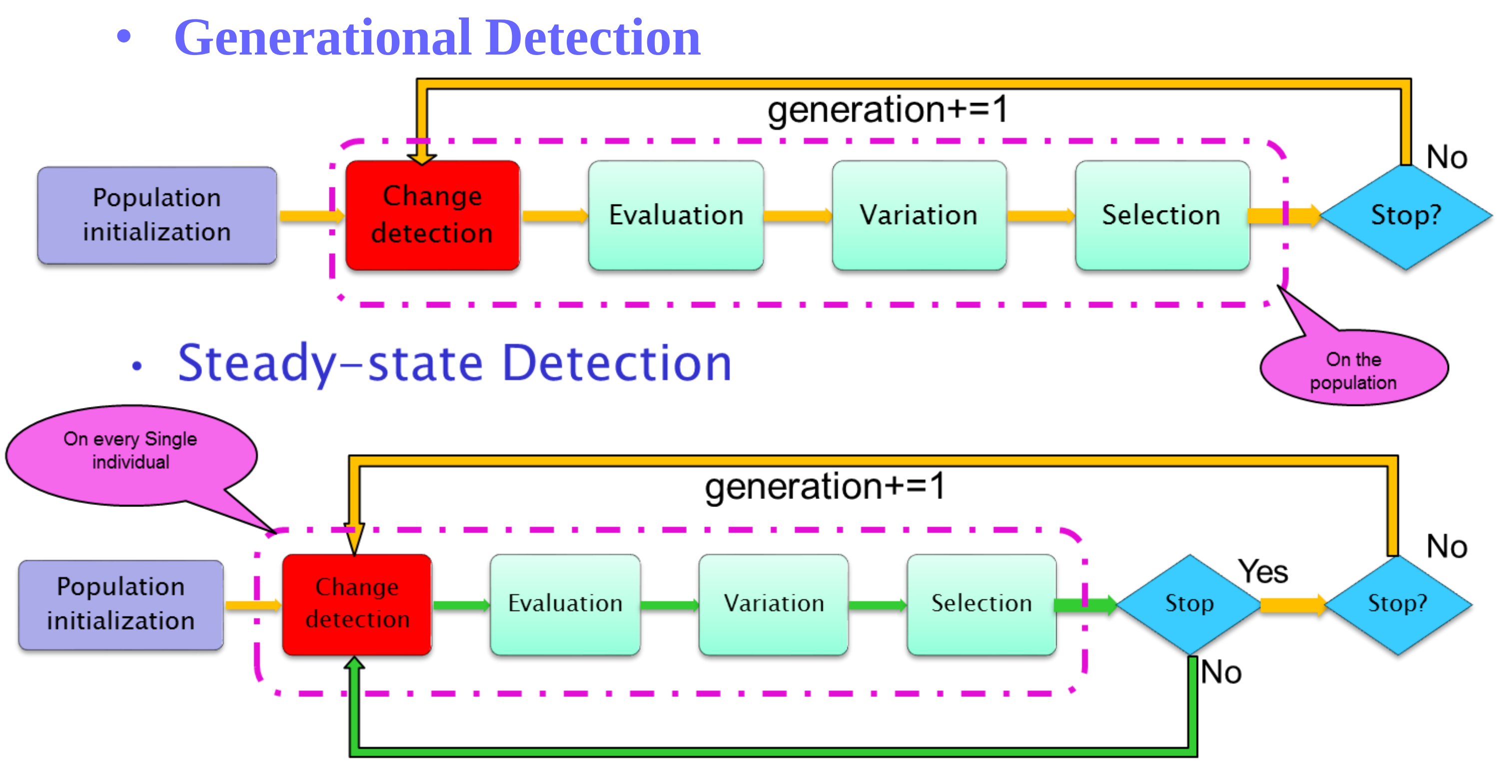
- Techniques:
  - ➢ Kalman filter (Muruganantham et al. 2016)
  - ➢ Population prediction strategy (Zhou et al. 2014)
  - ➢ Feed-forward prediction (Hatzakis & Wallace 2006)
  - ➢ Directed search strategy (Wu et al. 2015)
  - ➢ Evolutionary gradient search (Koo et al. 2010)
  - ➢ Center and knee points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

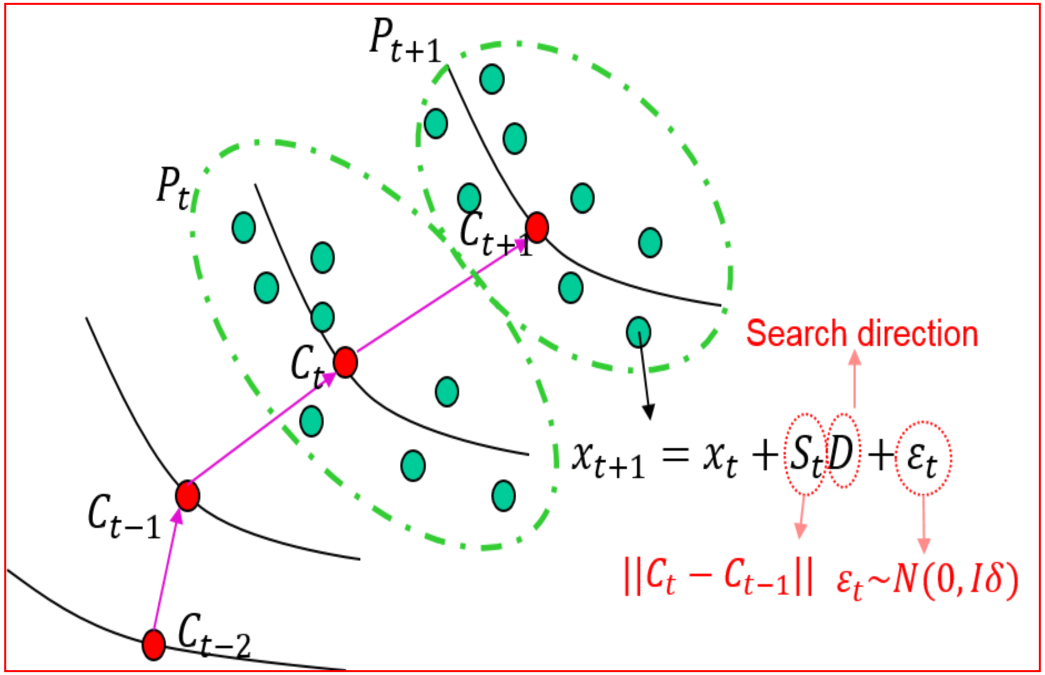
- Hybrid of steady-state and generational methods
- Steady-state detection in SGEA
  - Can detect a change in the middle of generation immediately
  - Rendering a fast follow-up action



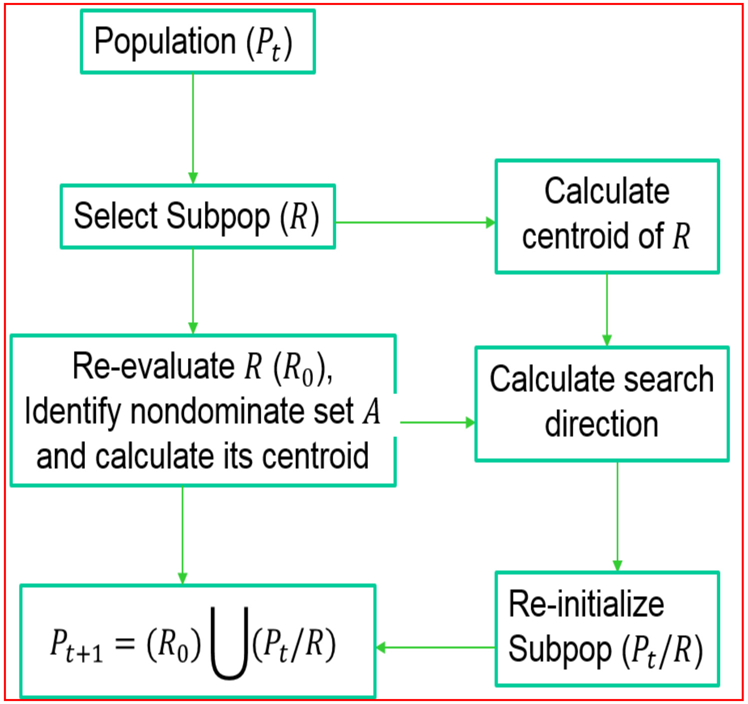
S. Jiang, S. Yang. A steady-state and generational evolutionary for dynamic multi-objective optimization. IEEE Transactions on Evolutionary Computation, 21(1): 65-82, 2017

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

Algorithm 1 Framework of SGEA

Input: N (population size)
Output: a series of approximated POFs
Create an initial parent population $P := \{x_1, \ldots, x_N\}$;
$(A, \overline{P}) :=$ `EnvironmentSelection`(P);
while *stopping criterion not met* **do**
    for $i := 1$ to N **do**
        if *change detected and not responded* **then**
            `ChangeResponse`();
        end if
        $y :=$ `GenerateOffspring`(P, A);
        $(P, A) :=$ `UpdatePopulation`(y);
    end for
    $(A, \overline{P}) :=$ `EnvironmentSelection`$(P \cup \overline{P})$;
    Set $P := \overline{P}$;
end while

---

steady-state

generational

Case Study: EA for Continuous DMOPs

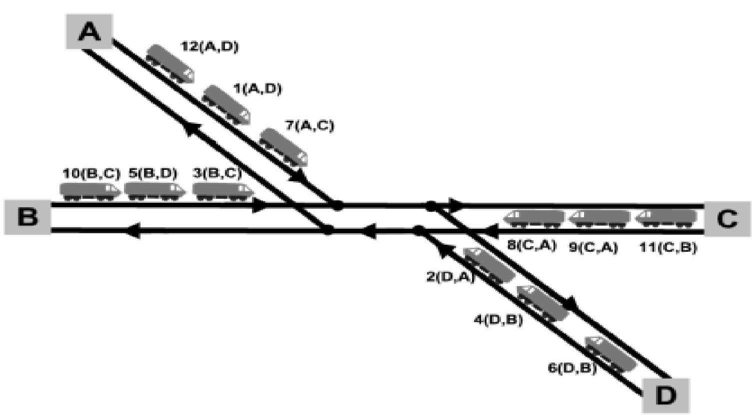
- Empirical study of SGEA
  - Test problems: FDA, dMOP, UDF, ...
  - Frequency of change: every 5, 10, 20 generations
  - Compared algorithms:
    - DNSGA-II: dynamic NSGAII (Deb et al. 2007)
    - dCOEA: Multi-population approach (Goh & Tan 2009)
    - PPS: population prediction strategy (Zhou et al. 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: ACO for DMOPs

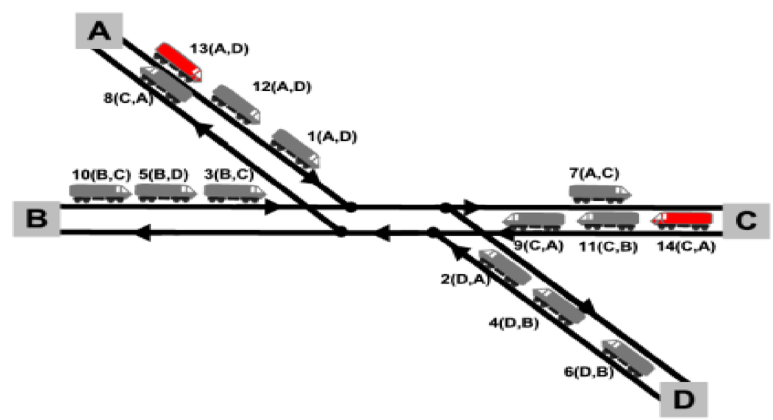
- ACO mimics the behaviour of ants searching for food
- ACO was first proposed for travelling salesman problems (TSPs) (Dorigo *et al.*, 1996)
- Generally, ACO is suitable for graph optimization problems, such as TSPs and vehicle routing problems (VRPs)
- The idea: let ants “walk” on the arcs of graph 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

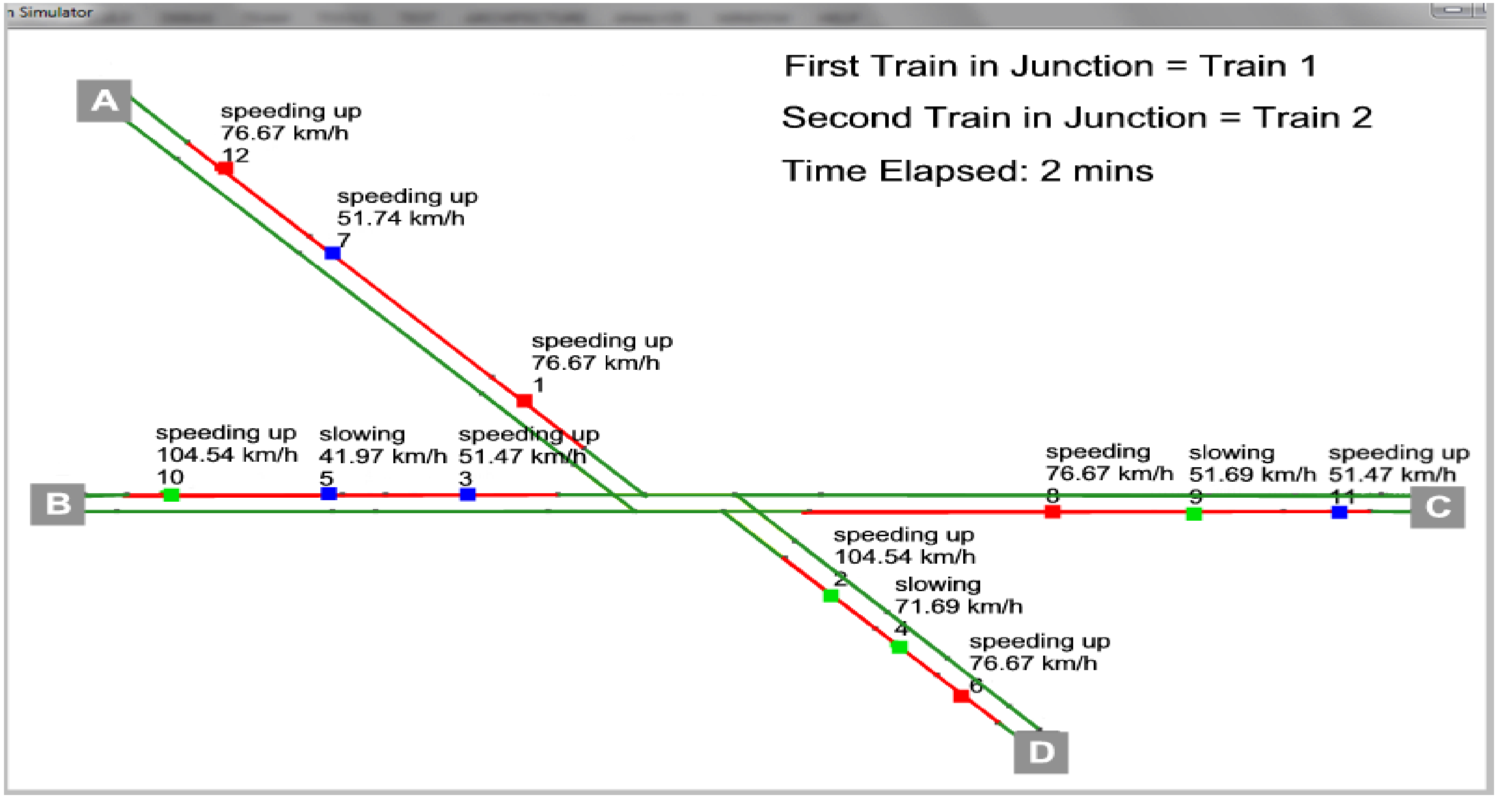


Junction after a change

J. Eaton, S. Yang, M. Gongora. Ant colony optimization for simulated dynamic multi-objective railway junction rescheduling. IEEE Trans Intell Transport Syst, 18(11): 2980-2992, 2017

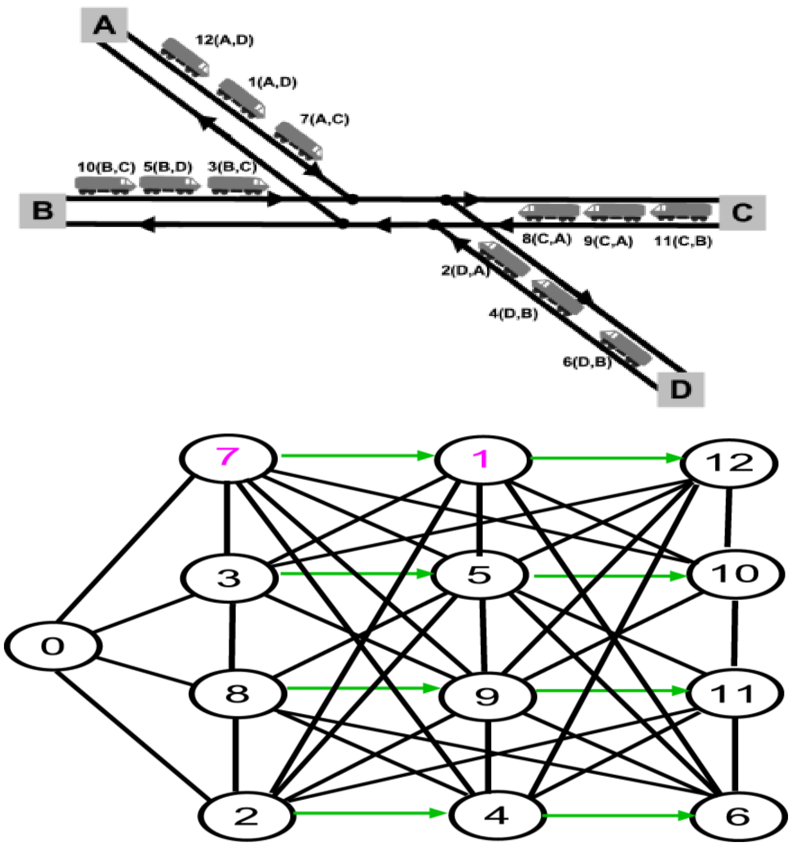
Case Study: ACO for DM-RJRP

- 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

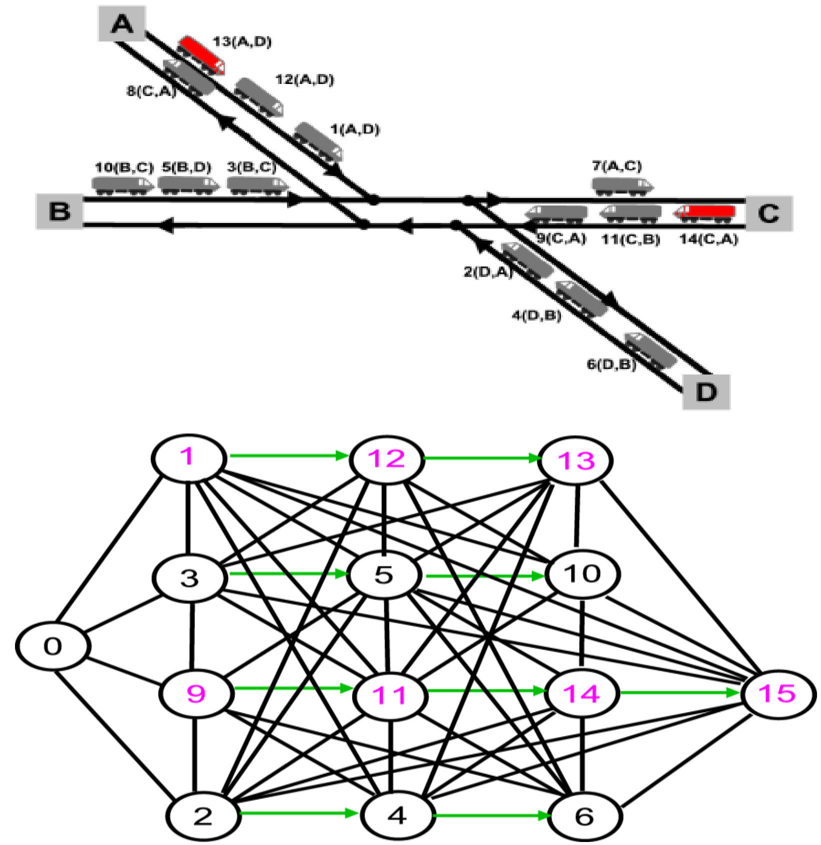


Case Study: ACO for DM-RJRP

- 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

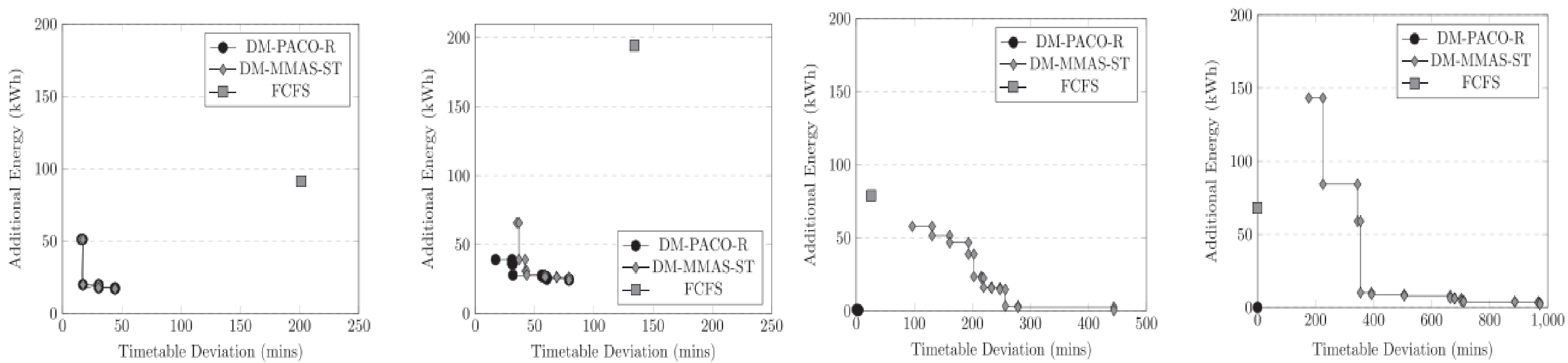
FOUR DIFFERENT VERSIONS OF THE DM-MMAS ALGORITHM

	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/

- 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 (2019). 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 for dynamic multi-objective optimization. IEEE Trans Evol Comput, 21(1): 65-82.
- 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.
- Q. Li, J. Zou, and S. Yang, J. Zheng, and G. Ruan. A predictive strategy based on special points for evolutionary dynamic multi-objective optimization. Soft Computing, 23(11): 3723-3739, 2019.

References-3

- Z. Liang, T. Wu, X. Ma, Z. Zhu, S. Yang. A dynamic multi-objective evolutionary algorithm based on decision variable classification. IEEE Trans Cybern, in press, 2020.
- 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.
- Y. Wu, Y. Jin, X. Liu (2015). A directed search strategy for evolutionary dynamic multiobjective optimization. Soft Computing 19(11): 3221-3235.

References-4

- 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.
- A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, E. Tsang (2007). Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization. Evolutionary Multi-Criterion Optimization, 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 Evolutionary Computation, 44: 247-259.