Evolutionary Computation for Dynamic Optimization Problems: Case Studies & Advanced Topics

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

Recall: EC for DOPs

- EC for DOPs: Important and young research area
- Approaches to enhance EC for DOPs:
  - Memory: store and reuse useful information
  - Diversity: handle convergence directly
  - Multi-population: co-operate sub-populations
  - Adaptive: adapt generators and parameters
  - Prediction: predict changes and take actions in advance
  - Hybrid methods

- M. Mavrovouniotis, C. Li, and S. Yang. A survey of swarm intelligence for dynamic optimization: Algorithms and applications. Swarm and Evolutionary Computation, 33: 1-17, April 2017
- T. T. Nguyen, S. Yang, and J. Branke. Evolutionary dynamic optimization: A survey of the state of the art. Swarm and Evolutionary Computation, 6: 1-24, October 2012

Outline of the Lecture

- Part II: Case studies
  - Population-Based Incremental Learning (PBIL) for binary DOPs
  - Particle swarm optimization (PSO) for continuous DOPs
  - EC for combinatorial DOPs
    - Genetic algorithms (GAs) for dynamic routing in MANETs
    - Ant colony optimization (ACO) for combinatorial DOPs
- Part III: Advanced Topics
- Summary

Case Study: Population-Based Incremental Learning

- PBIL: proposed by Baluja in 1994

  Initialize the probability vector $\vec{P}^0 := \vec{0.5}$
  repeat
      Generate a population of samples by $\vec{P}^t$
      Evaluate and denote the best sample by $\vec{B}^t$
      Learn $\vec{P}^t$ toward $\vec{B}^t$ by Eq. (1)
  until *terminated* = **true**

- Learning rule:

$$P_i^{t+1} := (1-\alpha) * P_i^t + \alpha * B_i^t, \quad i = \{1, \ldots, l\} \quad (1)$$

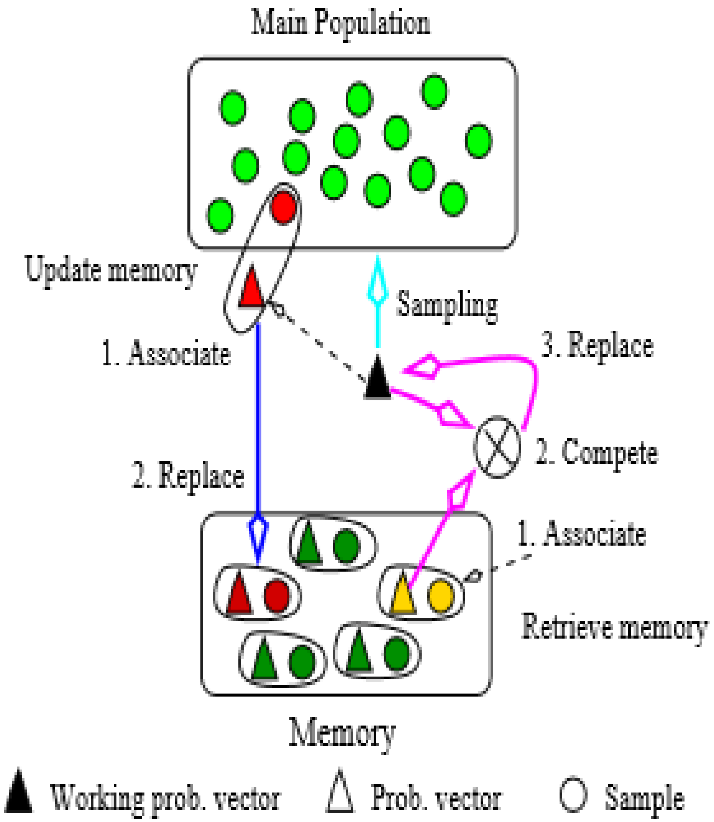
  where α is the learning rate

- Sampling a solution $\vec{S}$ by $\vec{P}^t$:
  For each locus i, if $rand(0,1) < P_i^t$, $S_i = 1$; otherwise, $S_i = 0$

S. Baluja, "Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning," Carnegie Mellon Univ., Tech. Rep. CMU-CS-94-163, 1994.

Case Study: PBIL with Associative Memory for DOPs

Idea: Uses a memory to store <$\vec{P}$, S> pairs



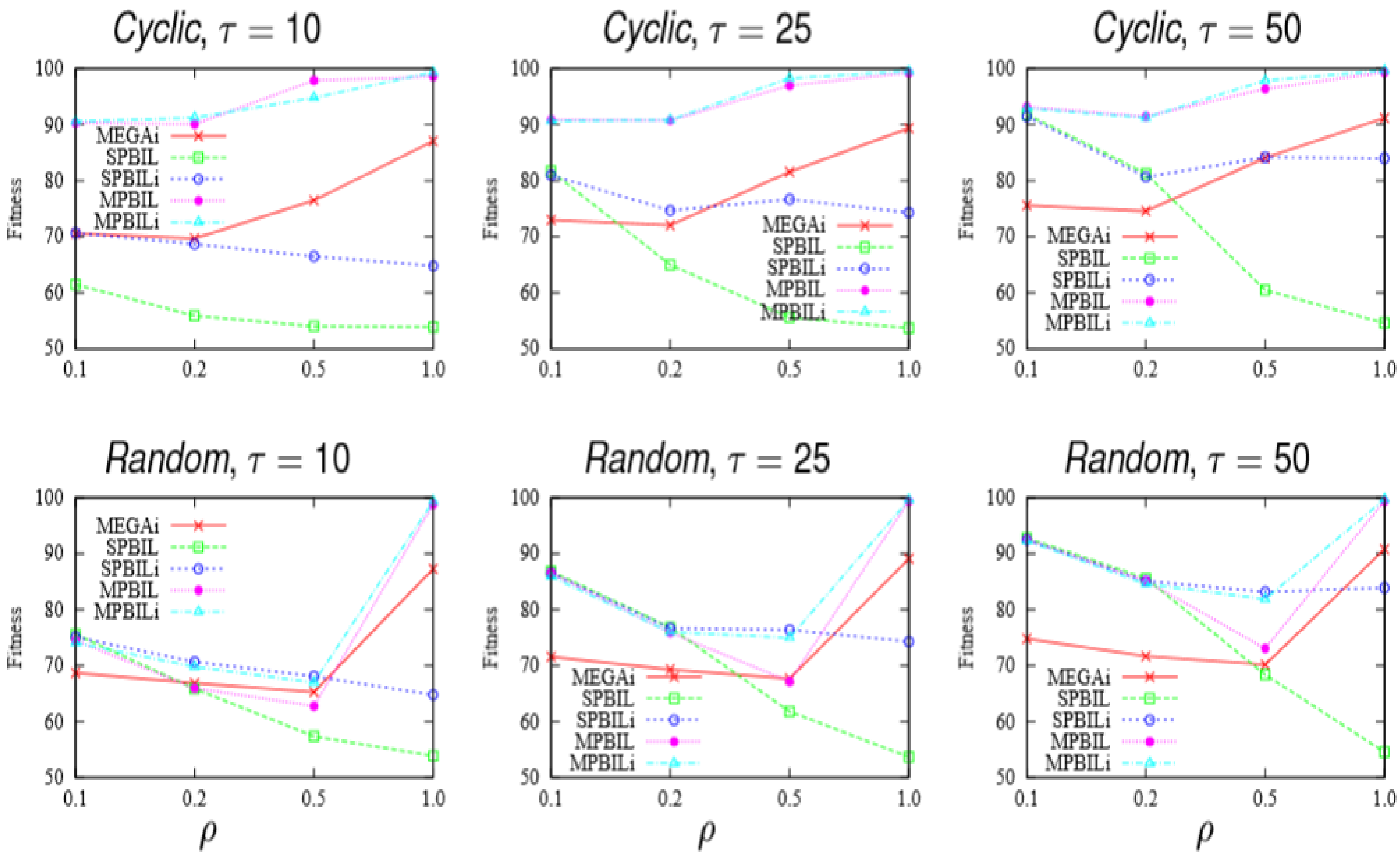
- Updated in a dynamic time pattern

$$t_M = t + rand(5, 10)$$

- Using the most similar strategy
  - Find memory sample closest to best pop sample
  - If the best pop sample is fitter, swap them and associated $\vec{P}$'s
- Re-evaluated every iteration
  - If change detected, best memory $\vec{P}_M$ competes with working $\vec{P}$

S. Yang. Population-based incremental learning with memory scheme for changing environments. Proceedings of the 2005 Genetic and Evolutionary Computation Conference, Vol. 1, pp. 711-718, 2005

PBIL with Associative Memory for DOPs: Results



- Associative memory efficiently improves PBIL's performance

Case Study: PSO for Continuous DOPs

- Developed by Kennedy and Eberhart (1995)
- A population based optimization technique inspired by social behaviour of bird flocking or fish schooling
- Swarm members can profit from their own discovery and previous experience of all other members of the school



Kennedy, J. and Eberhart, R.: Particle Swarm Optimization. Proceedings of the Fourth IEEE International Conference on Neural Networks, Perth, Australia. IEEE Service Center 1942-1948, 1995.

Particle Swarm Optimization (PSO)

- PSO consists of a **swarm** of **particles**
- Each particle resides at a **position** in the search space and flies over the search space with a certain **velocity**
- The **velocity** of each particle is influenced by
  - **Momentum:** maintaining previous velocity it has travelled so far
  - **Cognitive component:** returning to the best position visited so far
  - **Social component:** moving to the best position found by neighbors so far

$$\vec{v}_i \leftarrow \overbrace{\vec{v}_i}^{\text{Momentum}} + \overbrace{\vec{U}(0,\varphi_1)\otimes(\vec{p}_i - \vec{x}_i)}^{\text{Cognitive component}} + \overbrace{\vec{U}(0,\varphi_2)\otimes(\vec{g}_i - \vec{x}_i)}^{\text{Social component}}$$

$$\vec{x}_i \leftarrow \vec{x}_i + \vec{v}_i$$

- Eventually the swarm will converge to optimal positions

PSO for Continuous DOPs: Issues

- Recently, PSO has been applied for continuous DOPs
- Two aspects to consider:
  - Outdated memory. Two solutions:
    - Simply set *pbest* to the current position
    - Reevaluate *pbest* and reset it to current position if it is worse than the current position
  - Diversity loss. Three solutions:
    - Introduce diversity after a change
    - Maintain diversity during the run
    - Use multi-swarms

Multi-swarm PSO for DOPs

- Aim: To maintain multiple swarms on different peaks
- Key questions:
  - How to guide particles to different promising sub-regions?
  - How to determine the proper number of sub-swarms?
  - How to calculate the search area of each sub-swarm?
  - How to create sub-swarms?
- Algorithms:
  - Kennedy's k-means clustering algorithm
  - Brits's nbest PSO algorithm and niching PSO (NichePSO)
  - Parrott and Li's speciation based PSO (SPSO)
  - Blackwell and Branke's charged PSO (mCPSO) and quantum swarm optimization (mQSO)

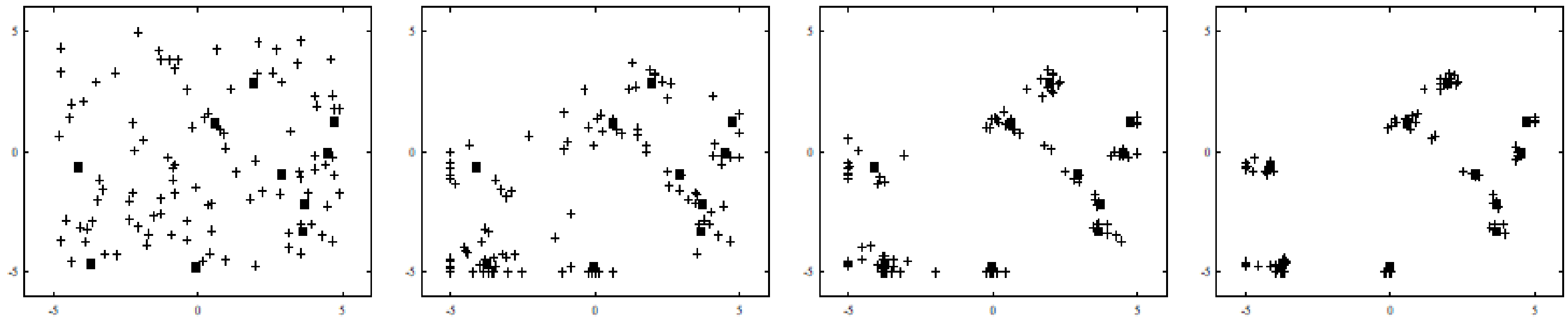
- T. Blackwell and J. Branke. Multiswarms, exclusion, and anti-convergence in dynamic environments. IEEE Trans Evol Comput, 10(4): 459-472, Aug. 2006.
- D. Parrott and X. Li. Locating and tracking multiple dynamic optima by a particle swarm model using speciation. IEEE Trans. Evol. Comput., 10(4): 440-458, Aug. 2006.

Multi-swarm PSO for DOPs

- Limitations of the above algorithms:
  - The number of sub-swarms is predefined (*k*-means PSO, mCPSO, and mQSO)
  - The search radius of each sub-swarm must be given by experimental experience (SPSO, mCPSO, and mQSO)
  - Simply create sub-swarms without analysing the population distribution (NichePSO and SPSO)
- Problems might be caused by the above algorithms:
  - There may be improper number of sub-swarms
  - One sub-swarm might cover more than one peak
  - One peak might be surrounded by more than one sub-swarm

Clustering PSO (CPSO) for DOPs

- Training: Move particles toward different regions



(a) initial stage (b) iteration 3 (c) iteration 6 (d) iteration 9

- Clustering sub-swarms: Single linkage hierarchical clustering
- Local search: Each sub-swarm searches one peak quickly
- Overlapping and convergence check stretagy
- Strategies to response to changes

- S. Yang and C. Li. A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments. IEEE Transactions on Evolutionary Computation, 14(6): 959-974, Dec. 2010
- C. Li and S. Yang. A clustering particle swarm optimizer for dynamic optimization. Proceedings of the 2009 IEEE Congress on Evolutionary Computation, pp. 439-446, 2009

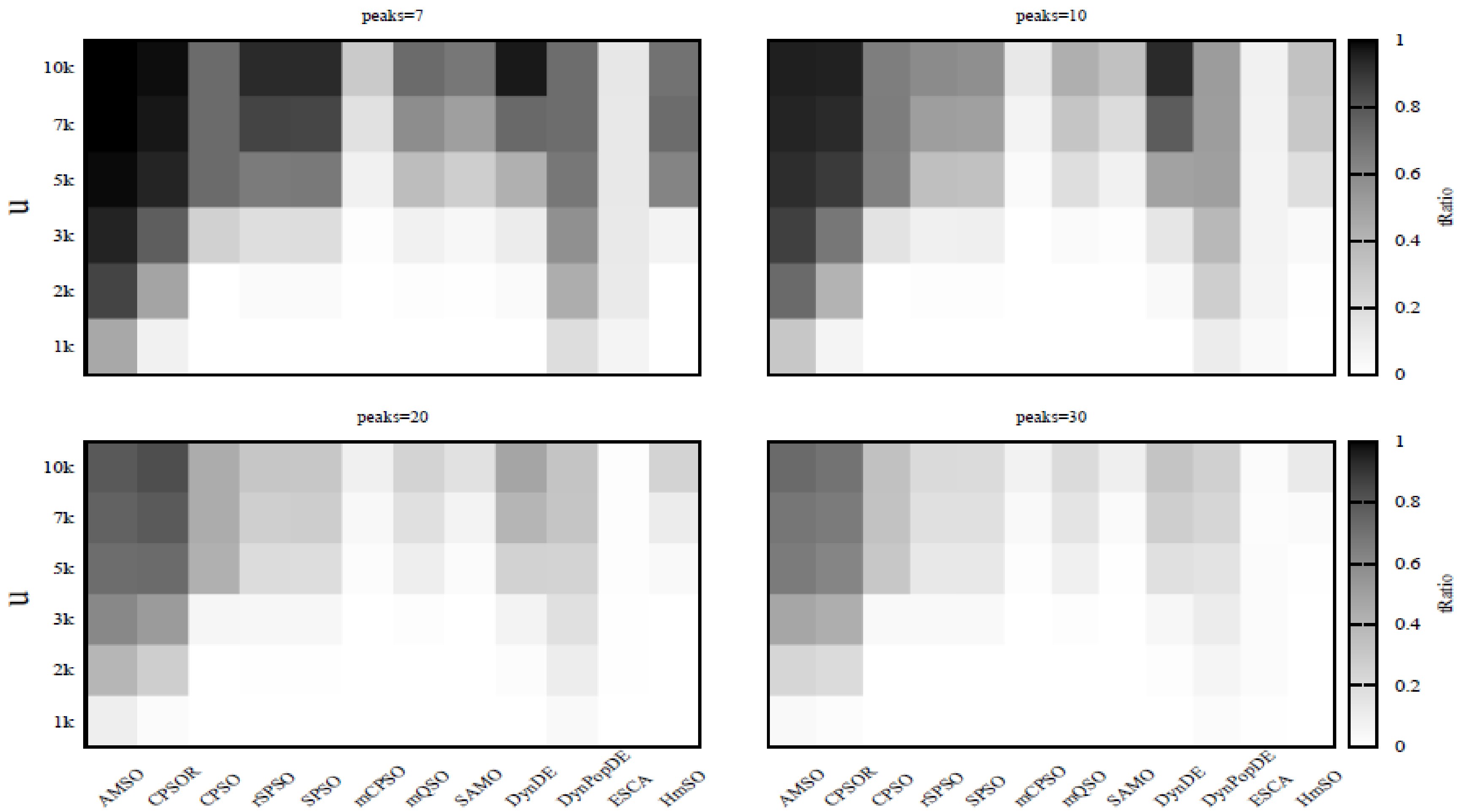
Adaptive Multi-Swarm Optimizer (AMSO)

- Recently, a framework of multi-population approaches
  - Use single linkage hierarchical clustering to create populations
  - Each population will search one peak in the fitness landscape
  - An overcrowding scheme to remove unnecessary populations
  - A special rule to decide proper moments to increase diversity without change detection
  - An adaptive method to create a proper number of populations needed

- C. Li and S. Yang. A general framework of multi-population methods with clustering in undetectable dynamic environments. IEEE Transactions on Evolutionary Computation, 16(4): 556-577, August 2012
- C. Li, S. Yang, and M. Yang. An adaptive multi-swarm optimizer for dynamic optimization problems. Evolutionary Computation, 22(4): 559-594, Winter 2014
- C. Li, T. T. Nguyen, M. Yang, M. Mavrovouniotis, and S. Yang. An adaptive multi-population framework for locating and tracking multiple optima. IEEE Transactions on Evolutionary Computation, 20(4):590-605, 2016

Multi-Swarm PSO for DOPs: Results

Average tracking ratio on MPB with different change frequencies

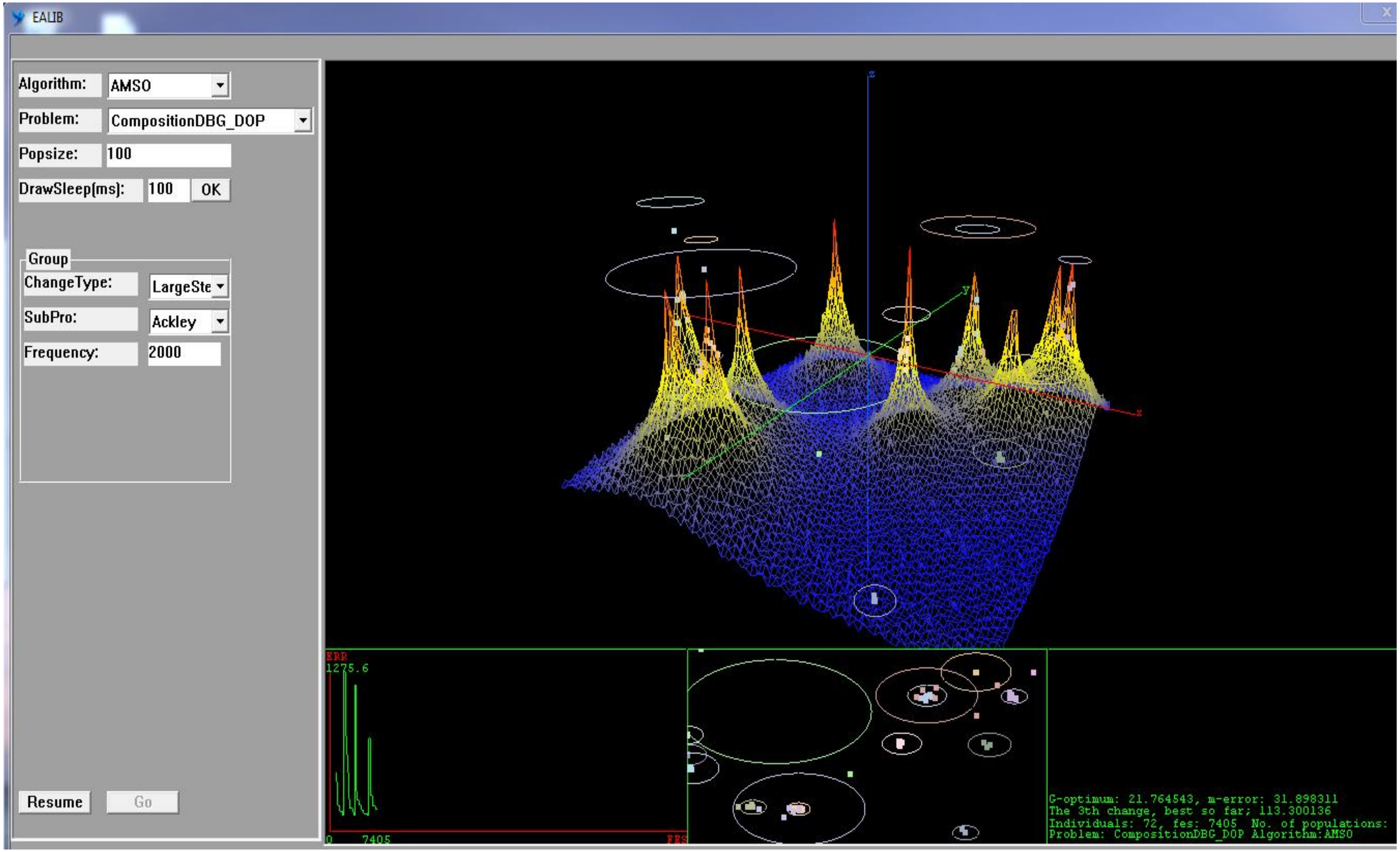


Multi-Swarm PSO for DOPs: Results

- The offline error (Eoffline) and best-before-change error (EBBC) on the MPB with changing number of peaks

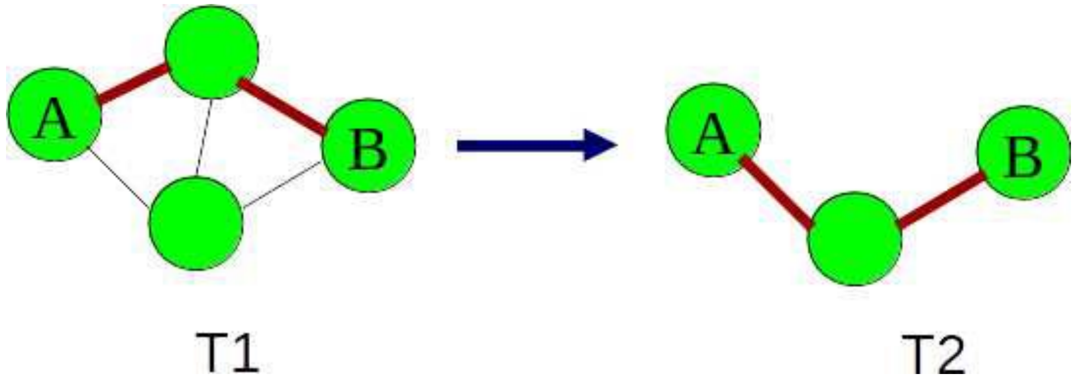
	Error	AMSO	CPSOR	CPSO	rSPSO	SPSO	mCPSO	mQSO	SAMO	DynDE	DynPopDE	ESCA	HmSO	SOS
Var1	$E_{offline}$	**2.3** ±0.25	2.8^{W} ±0.17	4^{W} ±0.28	6^{W} ±0.55	5.9^{W} ±0.58	7.8^{W} ±0.62	4.5^{W} ±0.27	3.5^{W} ±0.24	3.6^{W} ±0.38	3.7^{W} ±0.26	13^{W} ±1.3	4.5^{W} ±0.19	11^{W} ±3.2
	E_{BBC}	**1.5**	1.6^{W}	1.9^{W}	5.2^{W}	5.1^{W}	7^{W}	3.7^{W}	2.8^{W}	3^{W}	3.1^{W}	12^{W}	3.5^{W}	9.7^{W}
Var2	$E_{offline}$	**2.9** ±0.74	3.3^{W} ±0.63	5^{W} ±1	4.6^{W} ±0.65	4.9^{W} ±0.67	7.3^{W} ±0.89	4.4^{W} ±0.92	4^{W} ±0.68	3.5^{W} ±0.81	4.2^{W} ±0.75	13^{W} ±1.2	5.4^{W} ±0.69	9.4^{W} ±4.3
	E_{BBC}	2	$\mathbf{1.9}^{t}$	2.6^{W}	3.7^{W}	3.9^{W}	6.3^{W}	3.4^{W}	3.1^{W}	2.9^{W}	3.6^{W}	13^{W}	4^{W}	8.4^{W}
Var3	$E_{offline}$	**2.7** ±0.45	2.9^{W} ±0.29	4.5^{W} ±0.36	4.9^{W} ±0.68	4.8^{W} ±0.66	7.4^{W} ±0.98	4.1^{W} ±0.55	3.7^{W} ±0.32	3.4^{W} ±0.5	4.8^{W} ±0.59	13^{W} ±2	5.3^{W} ±0.4	9.6^{W} ±3.5
	E_{BBC}	1.7	$\mathbf{1.6}^{t}$	2.2^{W}	3.9^{W}	3.7^{W}	6.4^{W}	3.3^{W}	2.8^{W}	2.7^{W}	4.1^{W}	12^{W}	4.1^{W}	8.5^{W}

Multi-swarm PSO for DOPs: Demo



Case Study: GAs for Dynamic Routing in MANETs

- More and more mobile wireless networks, e.g., mobile ad hoc networks (MANETs), wireless mesh networks (WMNs)
- MANET: a wireless network set up temporarily without a wired infrastructure (routers, switches, servers, cables, access points, etc.)
- It is very suitable for disaster rescue and recovery battlefield communication, etc
- An important feature in MANETs is the topology dynamics due to energy conservation and/or node mobility
  - Node mobility ⇒ Topology change



Dynamic Shortest Path Routing in MANETs

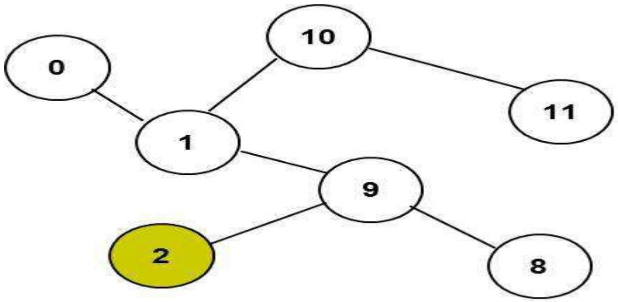
- Shortest path routing problem (SPRP) in a fixed network:
  - Find the shortest path between source and destination in a fixed topology
- In MANETs, the SPRP is a DOP
- Dynamic SPRP (DSPRP) in MANETs:
  - Find a series of shortest paths in a series of highly-related network topologies

Problem Model

- Given a MANET within a fixed geographical region, we model it by a undirected and connected topology graph G0(V0, E0)
  - V0 represents the set of wireless nodes (i.e., routers)
  - E0 represents the set of communication links connecting two neighboring routers within the radio transmission range
- Message transmission on a wireless communication link will incur remarkable delay and cost
- We model the network dynamics as follows:
  - For each change, a number of nodes are randomly selected to sleep or wake up based on their current status

Specialized GA for DSPRP in MANETs

- Path-oriented encoding:
  - A routing path is encoded by a string of integers that represent the IDs of nodes through which the path passes



0	1	9	2			

- Fitness function: The less the path cost, the better

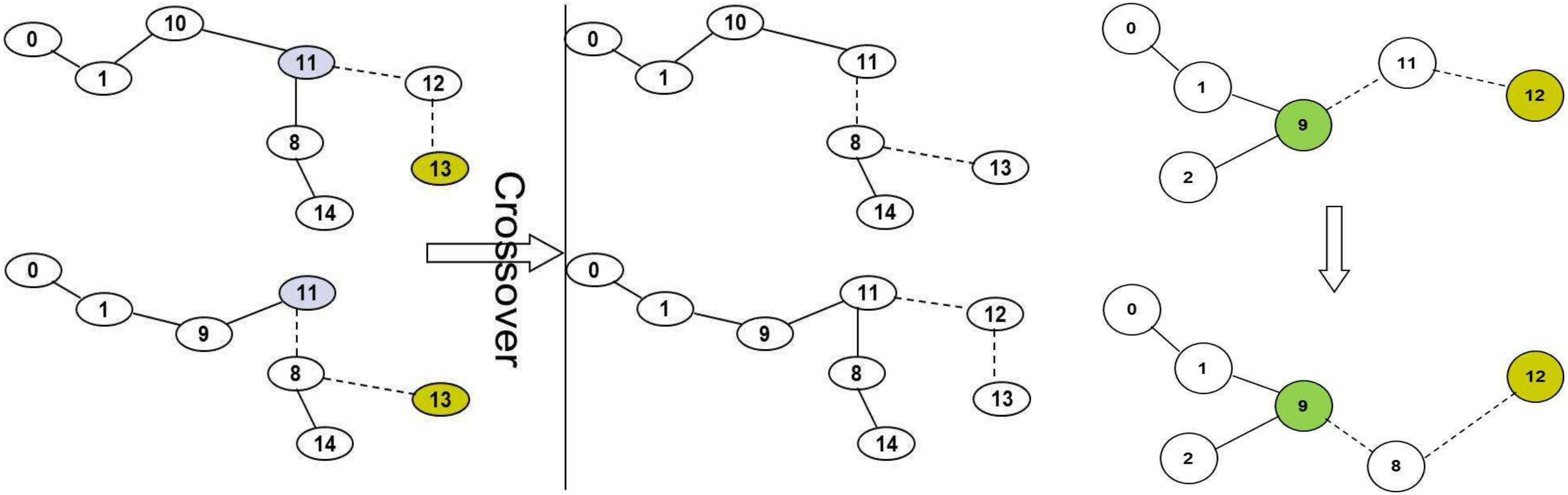
$$f(x) = [\sum_{l \in P(s,r)} c_l]^{-1}$$

where P(s, r) is the path from source s to destination r

- Pair-wise tournament selection

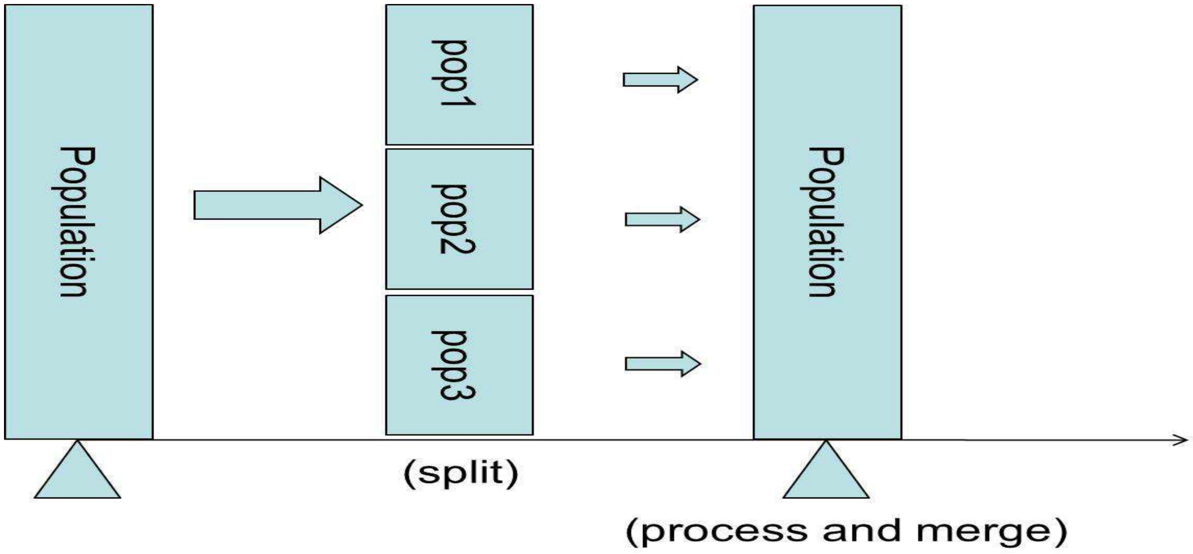
Specialized GA for DSPRP in MANETs

- Path-oriented crossover with repair:
  - Select two chromosomes with at least one common node
  - Randomly select one common node, denoted as v
  - Exchange the two sub-paths from v to r
- Path-oriented mutation with repair
  - Randomly select one node as the mutation point, denoted as v
  - Replace the sub-path from v to r by a new random one



Immigrants Enhanced Multi-Pop GA (iMPGA)

- Enhance specialized GA with immigrants and multi-pop
- For a given change interval R
  - At the first half interval, the whole population evolve
  - Then, split the whole population into 3 subpops (2 child populations and 1 parent population)
- Each generation, a small number of random immigrants are added into parent subpop, responsible for exploration
- When a change interval ends, we process 3 subpops separately and then merge them



Immigrants Enhanced Multi-Pop GA (iMPGA)

- Handling dynamic changes: When an environment change is detected, for each subpopulation:
  - If its best individual becomes infeasible, the whole subpop will be replaced by random solutions
  - If its best individual is feasible, only the infeasible individuals in the subpop will be replaced by random solutions

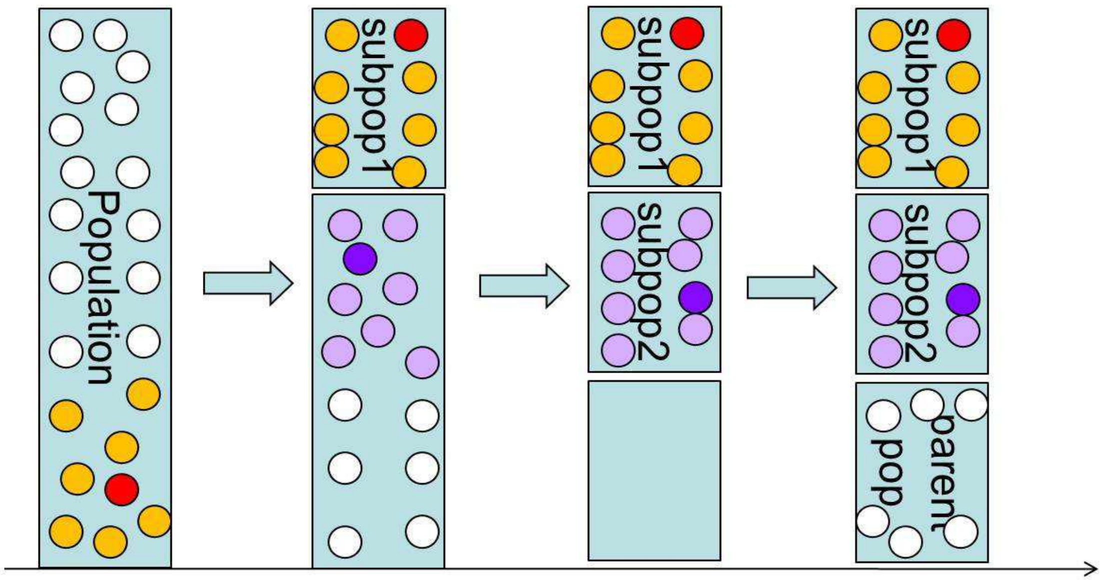
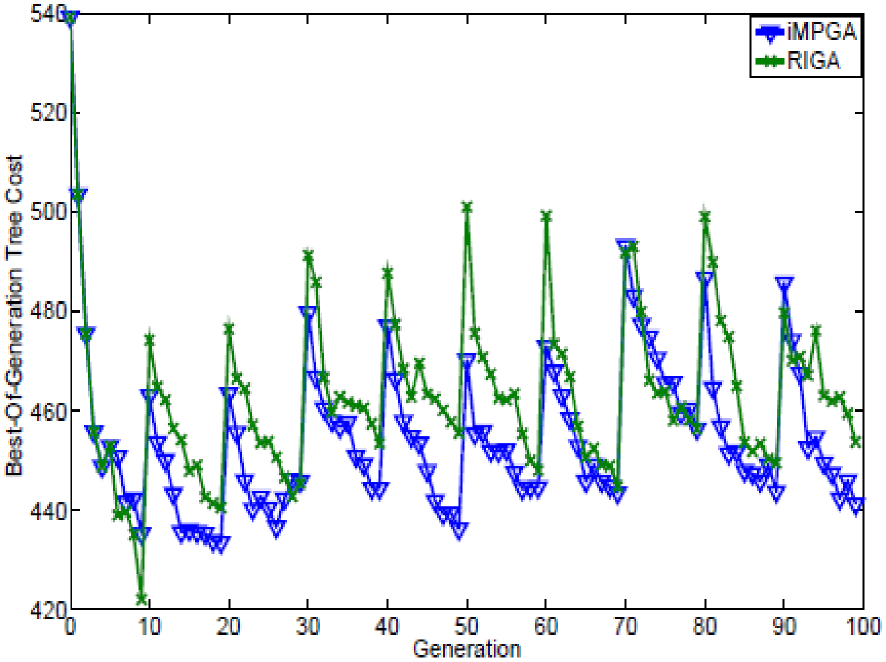
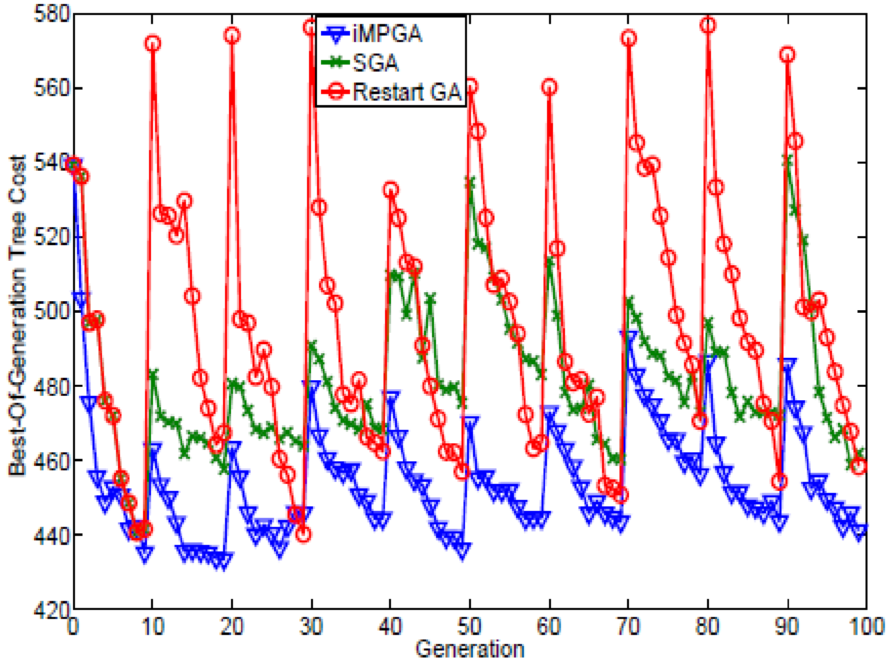


Illustration of the splitting

parent subpop: explore two child subpops: exploit

Immigrants Enhanced Multi-Pop GA (iMPGA)





- iMPGA outperforms traditional Gas
- Restart GA shows the worst performance
- iMPGA beats random immigrants GA (RIGA)
  - Multi-population improves GAs in handling dynamic environments

H. Cheng and S. Yang. Multi-population genetic algorithms with immigrants scheme for dynamic shortest path routing problems in mobile ad hoc networks. EvoApplications 2010: Applications of Evolutionary Computing, Part I, LNCS, vol. 6024, pp. 562-571, 2010

Immigrants and Memory Based GAs

- Enhance the above specialized GA with immigrants and memory schemes
  - Random immigrants GA (RIGA)
  - Elitism-based immigrants GA (EIGA)
  - Hybrid immigrants GA (HIGA)
  - Memory-enhanced GA (MEGA)
  - Memory-enhanced random immigrants GA (MRIGA)
  - Memory-based immigrants GA (MIGA)

Experimental Results

- Both immigrants and memory enhance GA's performance for the DSPRP in MANETs
- Immigrants schemes show their power in acyclic environments
- Memory related schemes work well in cyclic environments

Table: *t*-test results in acyclic environments

t-test Result	Topology Series #2			Topology Series #3		
Dynamics *R*	5	10	15	5	10	15
RIGA − *SGA*	s+	s+	s+	s+	s+	s+
EIGA − *SGA*	s+	s+	s+	s+	s+	s+
HIGA − *SGA*	s+	s+	s+	s+	s+	s+
RIGA − *HIGA*	+	+	+	−	−	−
EIGA − *HIGA*	−	−	−	−	−	−
MEGA − *HIGA*	s−	s−	s−	s−	s−	s−
MRIGA − *HIGA*	s−	s−	s−	s−	s−	s−
MIGA − *HIGA*	s−	s−	s−	s−	s−	s−

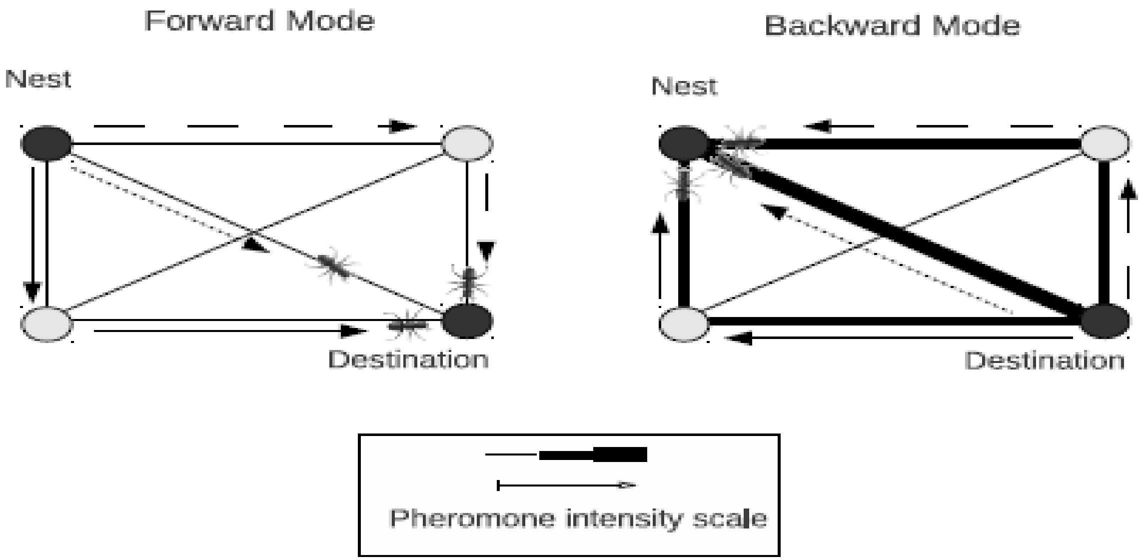
Table: *t*-test results in cyclic environments

t-test Result	Topology Series #1		
Environmental Dynamics *R*	5	10	15
MEGA − *SGA*	s+	s+	s+
MRIGA − *SGA*	s+	s+	s+
MIGA − *SGA*	s+	s+	s+
MEGA − *HIGA*	s+	s+	s+
MRIGA − *HIGA*	s+	s+	s+
MIGA − *HIGA*	s+	s+	s+

S. Yang, H. Cheng, and F. Wang. Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing problems in mobile ad hoc networks. IEEE Transactions on Systems, Man, and Cybernetics Part C: Applications and Reviews, 40(1): 52-63, Jan. 2010.

Case Study: ACO for Combinatorial DOPs

- Proposed by Dorigo et al. (1996)
- ACO mimics the behaviour of ants searching for food
- The idea: ants "walk" on the arcs of graph while "reading" and "writing" pheromones until they converge into a path
- The shorter the path the more pheromone deposited
- Standard ACO consists of two phases:
  - Forward mode: Construct solutions
  - Backward mode: Pheromone update

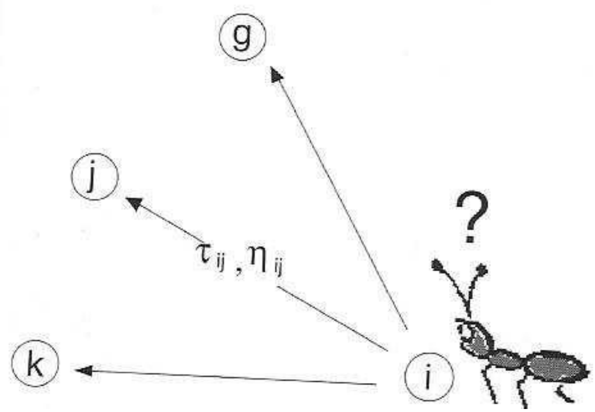


Forward Mode: Construct Solutions

- Ant k constructs a tour probabilistically

$$p_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha} [\eta_{ij}]^{\beta}}{\sum_{l \in N_i^k} [\tau_{il}]^{\alpha} [\eta_{il}]^{\beta}}, \text{ if } j \in N_i^k$$

  - τ_{ij} is the existing pheromone between cities *i* and *j*
  - η_{ij} is the heuristic information between cities *i* and *j*
  - N_i^k is the list of nearest unvisited cities of city *i*
  - α and β are constant parameters that determine the influence of τ and η, respectively



Backward Mode: Pheromone Update

- Ant k updates its pheromone trails
  - Deposit pheromone

$$\tau_{ij} \leftarrow \tau_{ij} + \Delta\tau_{ij}^{k}, \forall (i,j) \in T^{k}$$

    - T^k is the tour constructed by ant k
    - $\Delta\tau_{ij}^{k}$ is the amount of pheromone to be deposited

  - Evaporate pheromone

$$\tau_{ij} \leftarrow (1-\rho)\,\tau_{ij}, \forall (i,j)$$

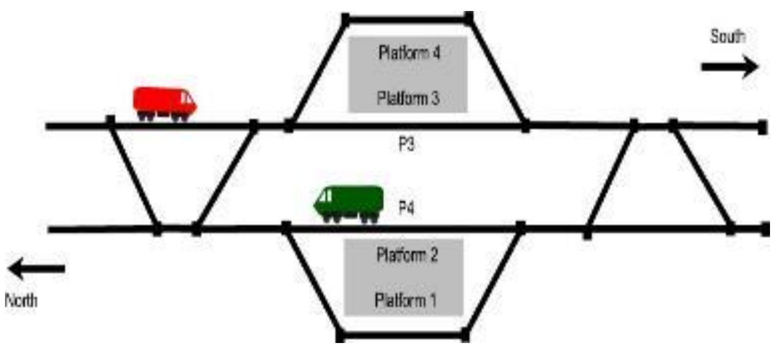
    - where ρ is the evaporation rate
    - Helps ants to "forget" bad decisions (poor solutions) made in the past (previous iteration): If an arc is not chosen by ants for a number of iterations, its associated pheromone value decreases exponentially

ACO for Combinatorial DOPs

- A train that arrives late at a station will miss its scheduled time slot and may have to be reallocated to a new platform
- Multiple trains may be delayed in succession, each new delay changes the problem
- Dynamic Railway Platform Reallocation Problem (DPRP) reallocates multiple successive delayed trains to new timeslots on railway platforms to minimise the ongoing delay in the system

Image source: https://en.wikipedia.org/wiki/Leicester_railway_station

Leicester Station Track Layout

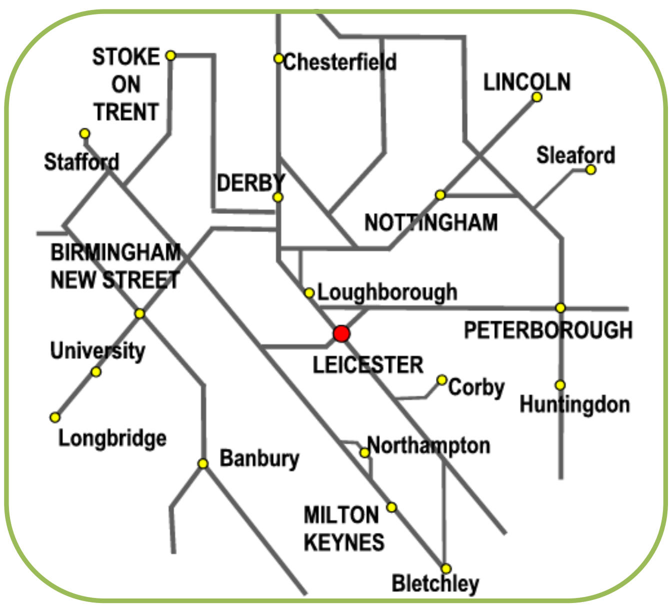


- We considered Leicester station
- A busy UK railway station with 4 bi-directional platforms and trains arriving from 4 different directions
- We consider the effect of the reallocation decisions not only at the station but also on the remainder of these trains' journey

Modelling the Problem

- The model was created from Network Rail's train schedule data from Integrated Train Planning System (ITPS)
- From this we extract details of the movement of trains through the station and the movement of all trains at each timing point on each train's route
- We consider timing points within 50 miles of Leicester station (225 timing points)

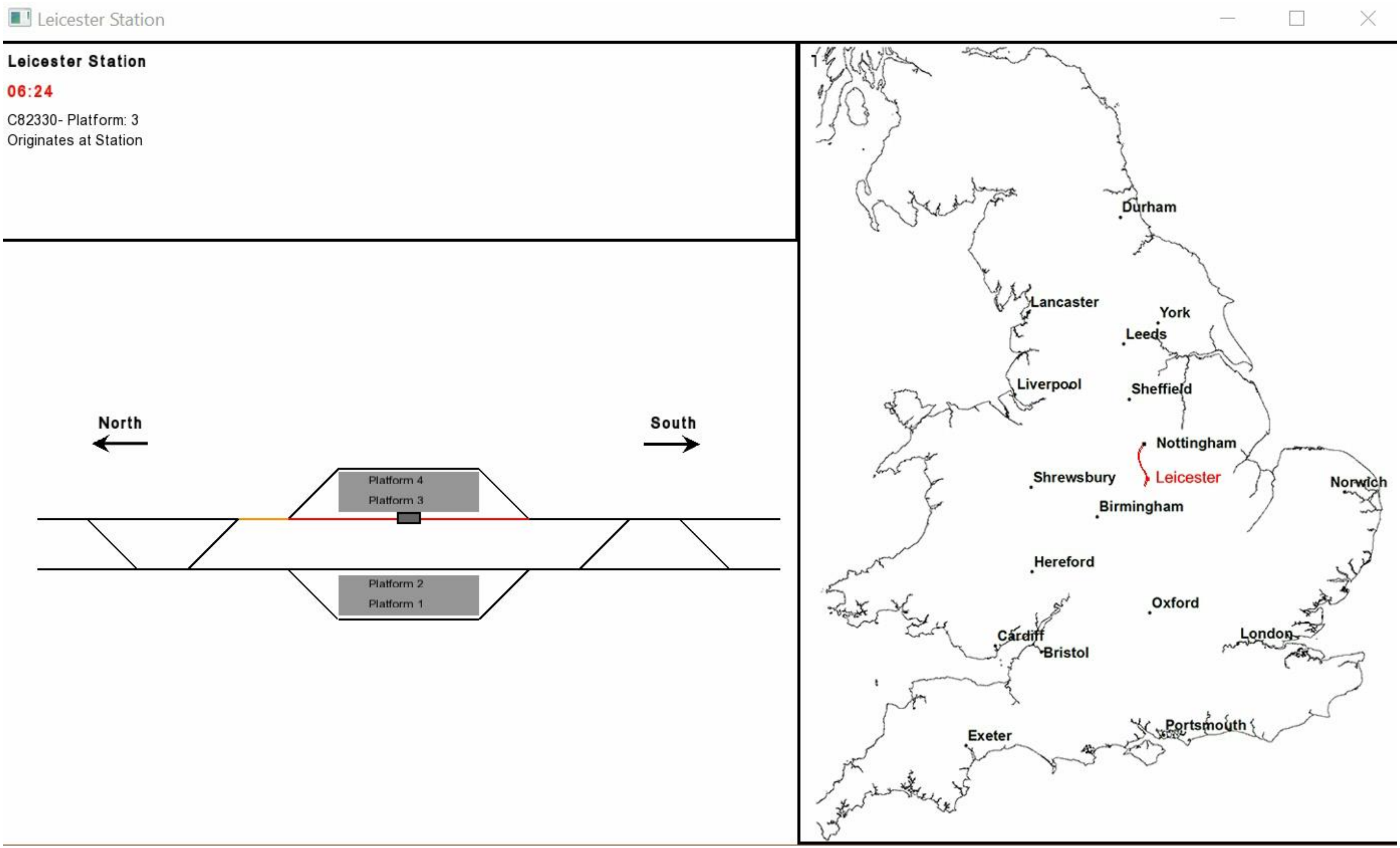
Some timing points in the problem



An example of the schedule feed data

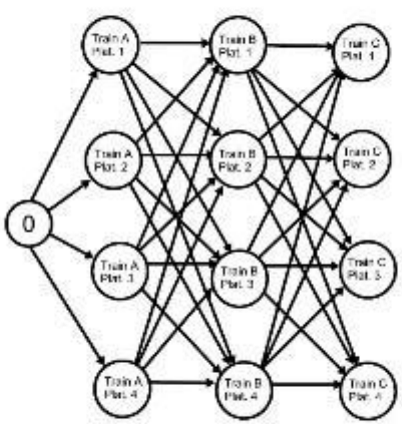
```
H06408,NA,NA,D,600,D,NA,28
LO,RATCFHH,NA,0434,NA,NA,NA,NA,NA,GL
LI,TRENTJ,NA,NA,NA,NA,0437,NA,NA,NA
LI,TRENT,NA,NA,NA,NA,0438,NA,NA,NA
LI,BESTNSJ,NA,NA,NA,NA,0445H,NA,NA,NA
LI,BESTONS,0447H,0516H,NA,NA,NA,NA,NA,NA
LI,BESTNSJ,NA,NA,NA,NA,0518H,NA,NA,NA
LI,TRENT,NA,NA,NA,NA,0523,NA,NA,NA
LI,TRENTJ,NA,NA,NA,NA,0525,NA,NA,GL
LI,RATCLFJ,NA,NA,NA,NA,0527,NA,NA,SL
LI,LOGHBRO,NA,NA,NA,NA,0536,USL,NA,NA
LI,SILEBYJ,NA,NA,NA,NA,0542,NA,NA,NA
LI,SYSTNSJ,NA,NA,NA,NA,0546,NA,NA,NA
LI,LESTER,NA,NA,NA,NA,0554H,3,SL,FL
LI,WGSTNNJ,NA,NA,NA,NA,0600,NA,NA,NA
LI,CROFTS,NA,NA,NA,NA,0606,NA,NA,NA
LI,HINCKLY,NA,NA,NA,NA,0616,NA,NA,NA
LI,NNTN,NA,NA,NA,NA,0626H,7,NA,SL
LI,AMNGTNJ,NA,NA,NA,NA,0637H,NA,SL,SL
LI,LCHTNJ,NA,NA,NA,NA,0646,NA,SL,SL
LI,RUGLYNJ,NA,NA,NA,NA,0653H,NA,SL,SL
LI,COLWICH,NA,NA,NA,NA,0706,NA,SL,FL
LI,MFDB,NA,NA,NA,NA,0708H,NA,NA,SL
LI,STAFTVJ,NA,NA,NA,NA,0715,NA,NA,NA
LI,STAFFRD,NA,NA,NA,NA,0717,5,SL,SL
LI,NTNB,NA,NA,NA,NA,0724,NA,SL,SL
```

Leicester Station Simulation

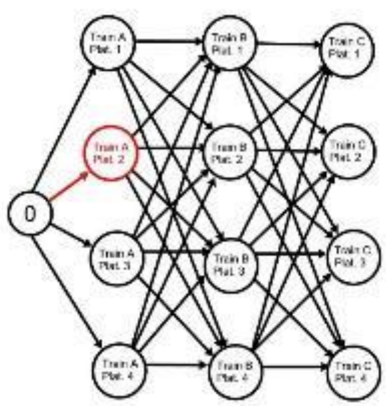


Max-Min Ant System (MMAS)

- In ACO ants communicate indirectly via pheromone trails
- We model the problem with a directed edge graph
- Ants choose next node based on pheromone trails and problem-specific heuristics

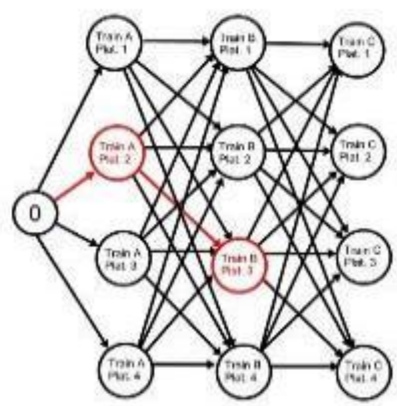


Each node in the graph represents a train and the platform to assign the train to



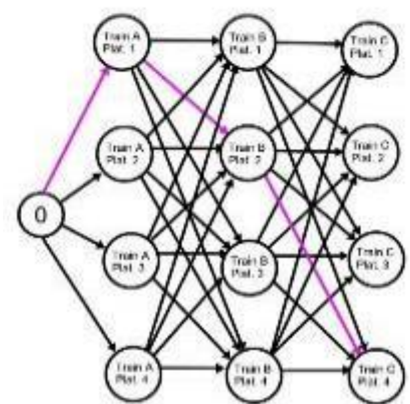
An ant starts on node 0
The ant chooses next node probabilistically

Ant Solution:
<Train A on Platform 2>



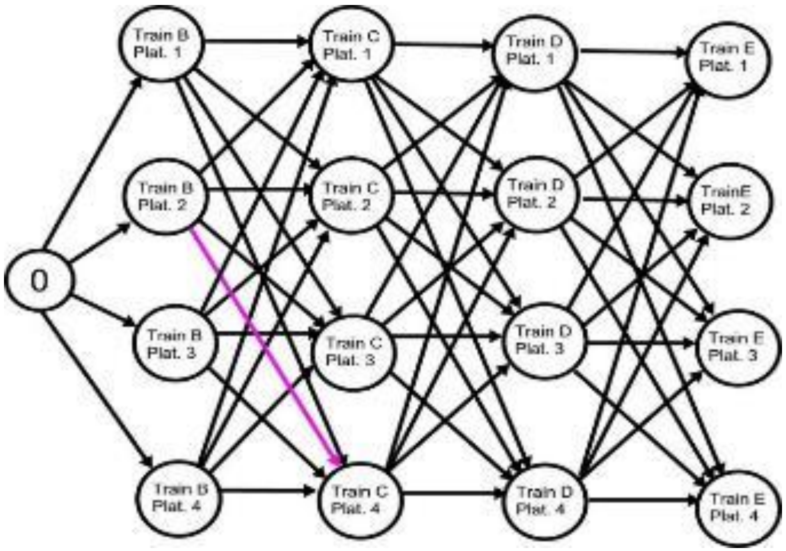
The ant now chooses the next train & platform

Ant Solution:
<Train A on Platform 2, Train B on Platform 3>



- After all ants have made a tour, all pheromone trails are evaporated
- Pheromone is laid down on the best ant's tour

Algorithm Design



After a Dynamic Change:

- More trains have arrived but some trains have passed through the station
- The graph is updated but pheromones are kept between changes to retain useful information from before change

Unnecessary Platform Reallocation:

- MMAS has no mechanism to persuade it against unnecessarily reallocating trains to platforms. To resolve this we:
  1. Add a heuristic based on the physical distance between platforms
  2. Introduce a best-so-far ant replacement scheme that discourages unnecessary reallocations of trains to new platforms

Image source: http://www.adelaidenow.com.au//

Comparison Algorithm

First Free Platform (FFP)

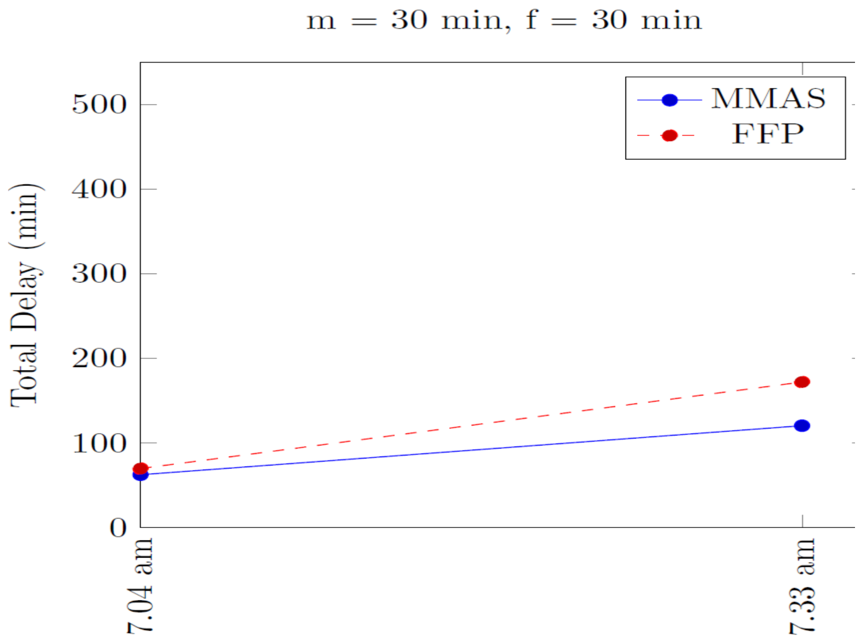
- Discussions with a Network Rail Station Master established that a technique often used to reallocate delayed trains to platforms is to find the first free platform as close as possible to the original platform
- We compared our MMAS algorithm to a heuristic using this principle

Modelling Dynamism

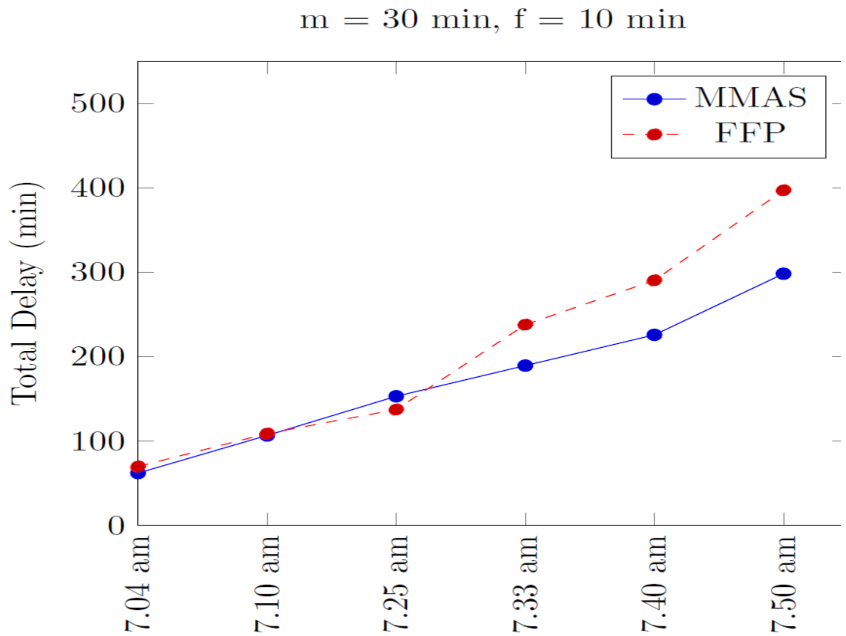
The **frequency of change f** is the time interval between delayed trains.
The **magnitude of change m** is how much the train is delayed by

In this investigation trains were delayed by 10, 20 and 30 min with gaps of 10, 20 and 30 min to give 9 different dynamic scenarios

Experimental Results



Low frequency, high magnitude changes



High frequency, high magnitude changes

Table 1. One-Sample Wilcoxon Signed Rank Test Results at 0.05 Significance

	m=30			m=20			m=10		
Algorithms	f=10	f=20	f=30	f=10	f=20	f=30	f=10	f=20	f=30
MMAS ⇔ FFP	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$	$s+$

J. Eaton and S. Yang. Railway platform reallocation after dynamic perturbations using ant colony optimisation. Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence, pp. 1-8, 2016

Part III: Advanced Topics

- EC for Dynamic Multi-objective Optimization
- EC for DOPs: Theoretical Development
- EC for DOPs: Challenging Issues
- EC for DOPs: Future Work

EC for Dynamic Multi-objective Optimization

- So far, mainly dynamic single-objective optimization
- Dynamic multi-objective optimization problems (DMOPs)
  - Even more challenging
- Recently, rising interest in studying EC for DMOPs
  - Farina et al. (2004) classified DMOPs by changes on Pareto optimal solutions
  - Goh & Tan (2009) proposed a competitive-cooperative coevolutionary algorithm for DMOPs
  - Zeng et al. (2006) proposed a dynamic orthogonal multi-objective EA (DOMOEA) to solve a DMOP with continuous decision variables
  - Zhang & Qian (2011) proposed an artificial immune system to solve constrained DMOPs
  - Jiang & Yang (2017a) proposed a new benchmark MDOP generator
  - Jiang & Yang (2017b) proposed a Steady-Generational EA for DMOPs
  - Ruan et al. (2017) analyzed the effect of diversity maintenance on prediction for DMOPs
  - Eaton et al. (2017) applied ACO for the dynamic multi-objective railway junction rescheduling problem

EC for DOPs: Theoretical Development

- So far, mainly empirical studies. Theoretical analysis has recently appeared
- Runtime analysis:
  - Stanhope & Daida (1999) first analyzed (1+1) EA on the dynamic bit matching problem (DBMP)
  - Droste (2002) analyzed first hitting time of (1+1) ES on the DBMP
  - Rohlfshagen et al. (2010) analyzed how the magnitude and speed of change may affect the performance of the (1+1) EA on two functions constructed from the XOR DOP generator

- S. Droste. Analysis of the (1+1) EA for a dynamically changing onemax-variant. CEC'02, pp. 55-60, 2002
- S.A. Stanhope, J.M. Daida. (1+1) genetic algorithm fitness dynamics in a changing environnements. Proceedings of the 1999 IEEE Congress on Evol Comput, vol. 3, pp. 1851-1858, 1999.
- P. Rohlfshagen, P.K. Lehre, X. Yao. Dynamic evolutionary optimisation: An analysis of frequency and magnitude of change. GECCO'09, pp. 1713-1720, 2009.

EC for DOPs: Theoretical Development

- Analysis of dynamic fitness landscape:
  - Branke et al. (2005) analyzed the changes of fitness landscape due to changes of the underlying problem instance
  - Richter (2010) analyzed the properties of spatio-temporal fitness landscapes constructed from Coupled Map Lattices (CML)
  - Tinos & Yang (2010, 2014) analyzed properties of the XOR DOP generator based on the dynamical system approach of the GA

- J. Branke, E. Salihoglu, S. Uyar. Towards an analysis of dynamic environments. GECCO'05, pp. 1433-1439, 2005.
- H. Richter. Evolutionary optimization and dynamic fitness landscapes: From reaction-diffusion systems to chaotic cml. Evolutionary Algorithms and Chaotic Systems, Springer, pp. 409-446, 2010.
- R. Tinos, S. Yang. An analysis of the XOR dynamic problem generator based on the dynamical system. PPSN XI, LNCS 6238, Part I, pp. 274-283, 2010.
- R. Tinos, S. Yang. Analysis of fitness landscape modifications in evolutionary dynamic optimization. Inform. Sci., 282: 214-236, 2014.

EC for DOPs: 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 DOPs
- Understanding the characteristics of DOPs:
  - What characteristics make DOPs easy or difficult?
  - Little work, needs much more effort
- Analysing the behaviour of EC methods for DOPs:
  - Requiring more theoretical analysis tools
  - Big question: Which EC methods for what DOPs?
- Real world applications:
  - How to model real-world DOPs?

EC for DOPs: 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
- EC for DMOPs: deserves much more effort
- Real world applications: also greatly needed
  - ➢ Fields: logistics, transport, MANETs, data streams, social networks, ...

Summary

- EC for DOPs: important area
  - The domain is still young and active
  - Many challenges to be taken
- 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