Dynamic Multi-objective Optimization: Introduction, Challenges, Applications and Future Directions

Mardé Helbig
School of ICT, Griffith University
Gold Coast Campus, Southport, QLD, Australia
m.helbig@griffith.edu.au

http://gecco-2021.sigevo.org/

GECCO '21 Companion, July 10–14, 2021, Lille, France

ACM ISBN 978-1-4503-8351-6/21/07…$15.00
https://doi.org/10.1145/3449726.3461413

1

Instructor

Mardé Helbig is a Senior Lecturer at the School of ICT and an associate member of the Institute for Integrated and Intelligent Systems at Griffith University. Her research focuses on solving dynamic multi-objective optimization (DMOO) problems using computational intelligence algorithms. She is the chair of the IEEE Computational Intelligence Society's (IEEE CIS) Chapters sub-committee, co-chair of the IEEE CIS Mentoring sub-committee and a member of the IEEE CIS Women in Computational Intelligence and Young Professionals sub-committees. She has received the 2018/2019 TW Kambule-NSTF: Emerging Researcher award for her research on DMOO.

https://experts.griffith.edu.au/25256-mard%C3%A9-helbig
https://www.griffith.edu.au/institute-integrated-intelligent-systems

2

Where in the World am I Located?

[19]

3

Final Version of Slides

Final version of slides will be available at:
https://experts.griffith.edu.au/25256-mard%C3%A9-helbig/publications

4

Course Agenda

1. Introduction:
   - ❖ Multi-objective optimization (MOO)
   - ❖ Dynamic multi-objective optimization (DMOO)
2. Analyzing algorithms' performance:
   - ❖ Benchmark functions
   - ❖ Performance measures
   - ❖ Analyzing the results
3. Real-world applications
4. Challenges in DMOO field
5. Emerging areas
6. Conclusions

5

Introduction
Multi-objective Optimization

- ❖ Imagine you want to buy a car
- ❖ Want to have as much comfort as possible and pay as little as possible
- ❖ More than one goal
- ❖ Multi-objective optimization problem

[1]

6

Introduction
Multi-objective Optimization

7

Introduction
Multi-objective Optimization

Goals are in conflict with one another:

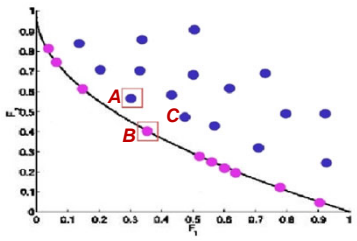
- ❖ Improving one objective leads to deterioration of another
- ❖ A single solution does not exist
- ❖ Need a new way to quantify the quality of solution

[1]

8

Introduction
Multi-objective Optimization

Pareto-dominance Approach



- Assume minimization
- *A* is better than *B* w.r.t. F_1, but worse than *B* w.r.t. F_2
- *B* is better than *A* w.r.t. F_2, but worse than *A* w.r.t. F_1
- Both deemed as equally good
- *A* and *B* are non-dominated solutions
- *B* is better than *C* w.r.t. both objectives
- *B* dominates *C* or *C* is dominated by *B*

9

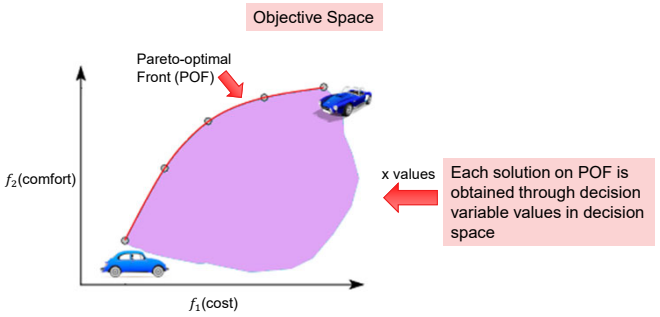
Introduction
Multi-objective Optimization

- Goal is to find a set of optimal trade-off solutions
- Work in 2 spaces:
  - Decision space: called the Pareto-optimal set (POS)
  - Objective space: called the Pareto-optimal front (POF)

[1]

10

Introduction
Multi-objective Optimization



11

Introduction
Multi-objective Optimization
Mathematical Formulation

minimise : $\mathbf{f}(\mathbf{x})$ ← Objective functions

subject to: $g_i(\mathbf{x}) \leq 0$, i=1,…,n_g ← Inequality constraints

$h_j(\mathbf{x}) = 0$, j=1,…,n_h ← Equality constraints

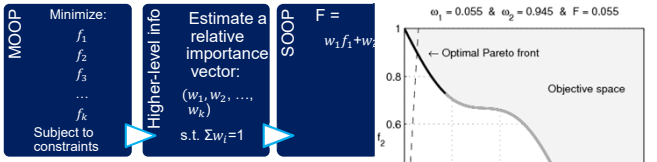
$\mathbf{x} \in [\mathbf{x}_{min}, \mathbf{x}_{max}]^{n_x}$ ← Boundary constraints

Decision variable space

12

Introduction
Multi-objective Optimization
Preference-based Approaches: Weighted Sum Approach

Any found solution is Pareto-optimal



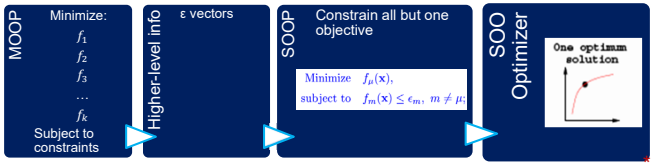
[2]

Problems with this approach:
- Choosing weights introduces another optimization problem
- Expect a lot of expert knowledge
- Obtain a single solution per run
- Absence of parallel search
- Cannot find solutions in non-convex area of POF

13

Introduction
Multi-objective Optimization
Preference-based Approaches – ε Constraint Approach

Can find any Pareto-optimal solution

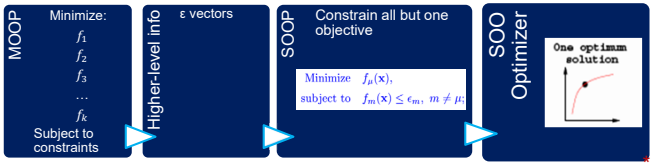


Problems with this approach:
- Need to know the relevant ε vectors
- Expect a lot of expert knowledge
- Obtain a single solution per run
- Absence of parallel search
- Non-uniformity in Pareto-optimal solutions

14

Introduction
Multi-objective Optimization
Preference-based Approaches – ε Constraint Approach

Can find any Pareto-optimal solution



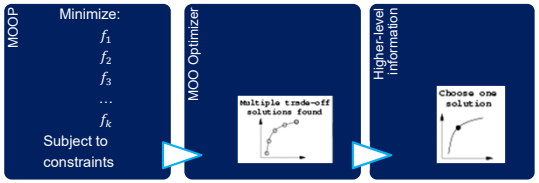
Problems with this approach:
- Need to know the relevant ε vectors
- Expect a lot of expert knowledge
- Obtain a single solution per run
- Absence of parallel search
- Non-uniformity in Pareto-optimal solutions

Play video

[3]

15

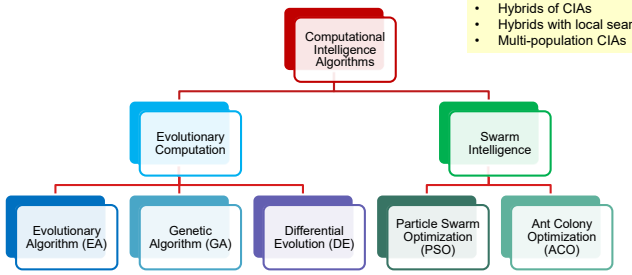
Introduction
Multi-objective Optimization
Population-based Approaches (CI Approaches)



Advantages of this approach:
- Population of entities
- Each entity is a possible solution – multiple solutions per run
- Can find solutions in non-convex area of POF
- Do not require expert knowledge to assign weights

16

Introduction
Multi-objective Optimization
Population-based Approaches (CI Approaches)



Also:
- Hybrids of CIAs
- Hybrids with local search
- Multi-population CIAs

For a review on CI algorithms for DMOO, refer to [8]

17

Introduction
Multi-objective Optimization
Population-based Approaches (CI Approaches)

Each entity is a possible solution

Algorithm 2: NSGA-II Algorithm

$t = 0$
Initialise parent population P_t
while *stopping condition has not been reached* **do**
  Produce offspring — Crossover — Mutation
  Combine parents and offspring in a combined population R
  Sort R according to Pareto-ranking
  Select individuals for the new population, P_{t+1}
  $t = t + 1$
end

Algorithm 3 Pareto ranking
1. set population P_r equal to the whole population P
2. set $i = 0$
3. while P_r is not empty
4. select individuals from P_r that are non-dominated
5. assign rank i to the selected individuals and remove them from P_r
6. increment i

18

Introduction
Multi-objective Optimization
Population-based Approaches (CI Approaches)

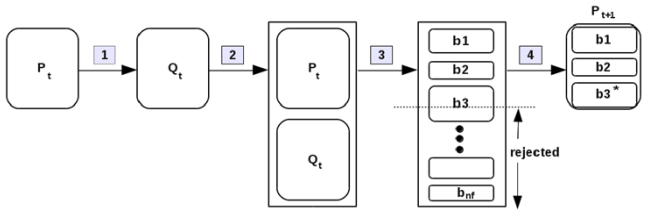


Figure 6.1: Steps of NSGA-II [4]

19

Introduction
Multi-objective Optimization
Population-based Approaches (CI Approaches)

Algorithm 2: NSGA-II Algorithm

$t = 0$
Initialise parent population P_t
while *stopping condition has not been reached* **do**
  Produce offspring — Crossover — Mutation
  Combine parents and offspring in a combined population R
  Sort R according to Pareto-ranking
  Select individuals for the new population, P_{t+1}
  $t = t + 1$
end

20

Introduction
Multi-objective Optimization
Population-based Approaches (CI Approaches)

Algorithm 3: OMOPSO Algorithm

```
t = 0
Initialise population P_t
Place leaders in ε-archive A_t
while stopping condition has not been reached do
    for each particle in swarm do
        Select leader
        Update particle's velocity and position
        Apply mutation
        Evaluate particle
        Update the particle's pbest
    end
    Update leaders
    Place leaders in A_t and update archive to produce A_{t+1}
    Calculate the crowding distance of the leaders
    t = t + 1
end
```


21

Introduction
Dynamic Multi-objective Optimization

22

Introduction
Dynamic Multi-objective Optimization

- Many problems in real life are dynamic
- Car example => may become a parent:
  - Priorities change: comfort less important, safety more
- Manufacturing example: Job scheduling of production plant:
  - Order's priority may change – more urgent job come in
  - Machine may break down

[1]

23

Introduction
Dynamic Multi-objective Optimization
Detecting a Change

1. Detect change with sentries
2. Divide population into 2 parts:
   - Some explore search space
   - Some exploit
   - Examples: Quantum PSO, Charged PSO

[4]

24

Introduction
Dynamic Multi-objective Optimization
Responding to Change

Need to deal with two major issues:

1. Outdated memory
2. Diversity loss

25

Introduction
Dynamic Multi-objective Optimization
Responding to Change: Outdated Memory

- Individuals converged to POF in previous environment
- Solutions may not be optimal anymore
- Do not just want to throw knowledge away - change may be small
- But attractors pull other entities to previous good solutions
- Need to address this issue:
  - ➡ Re-evaluate attractors and stored solutions (archive)

PSO

Re-evaluate *pbest*

Re-evaluate *nbest/gbest*

Re-evaluate *archive*

26

Introduction
Dynamic Multi-objective Optimization
Responding to Change: Diversity Loss

- Individuals converged to POF in previous environment
- Do not explore the search space anymore
- Need to address this issue:
  - ➡ Inject some diversity

PSO

Re-initialize % of particle

Which particles? *gbest*?

EC

Re-initialize % of individuals

Mutate % of individuals

Which one to use when?

27

Introduction
Dynamic Multi-objective Optimization
Population-based Approaches (CI Approaches)

Algorithm 2: NSGA-II Algorithm

```
1 t = 0
2 Initialise parent population P_t
3 while stopping condition has not been reached do
4 |   Produce offspring
5 |   Combine parents and offspring in a combined population R
6 |   Sort R according to Pareto-ranking
7 |   Select individuals for the new population, P_{t+1}
8 |   t = t + 1
9 end
```

Select sentries

1. Check for a change
2. If a change, respond:
   - Inject diversity
   - Check selection

28

Introduction
Dynamic Multi-objective Optimization
Population-based Approaches (CI Approaches)

Algorithm 3: OMOPSO Algorithm

```
1  t = 0
2  Initialise population P_t
3  Place leaders in ε-archive A_t
4  while stopping condition has not been reached do
5      for each particle in swarm do
6          Select leader
7          Update particle's velocity and position
8          Apply mutation
9          Evaluate particle
10         Update the particle's pbest
11     end
12     Update leaders
13     Place leaders in A_t and update archive to produce A_{t+1}
14     Calculate the crowding distance of the leaders
15     t = t + 1
16 end
```

1. Check for a change
2. If a change, respond:
   - Inject diversity
   - Update pbest
   - Update nbest/gbest
   - Update archive

Select sentries

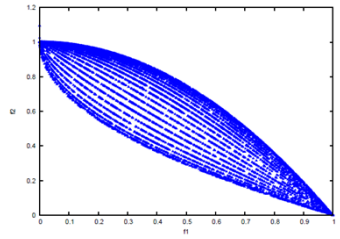
29

Analyzing Algorithms' Performance
Benchmarks

30

Dynamic Multi-objective Optimization
Benchmarks

- Objectives and/or constraints change over time
- Do not want to re-start algorithm
- Algorithm has to track the changing POS/POF over time



POF of dMOP2$_{iso}$ [5]

Each line is a POF for a specific environment

31

Dynamic Multi-objective Optimization
Benchmarks
Mathematical Formulation

minimise : $\mathbf{f}(\mathbf{x}, t)$ ← Objective functions

subject to: $g_i(\mathbf{x}, t) \leq 0, i=1,\ldots,n_g$ ← Inequality constraints

$h_j(\mathbf{x}, t) = 0, j=1,\ldots,n_h$ ← Equality constraints

$\mathbf{x} \in [\mathbf{x}_{min}, \mathbf{x}_{max}]^{n_x}$ ← Boundary constraints

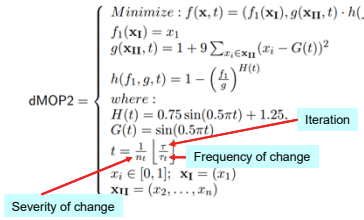
Decision variable space

32

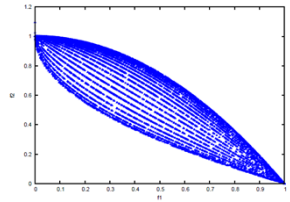
Dynamic Multi-objective Optimization
Benchmarks
DMOO Problem Types

Farina, Deb and Amato[] identified 4 main types of DMOO problems (DMOOPs):

	Static/Dynamic	
Type	**POS**	**POF**
I	Dynamic	Static
II	Dynamic	Dynamic
III	Static	Dynamic
IV	Static	Static



DMOO algorithm has to:
- Detect/determine a change
- Respond to a change



POF of dMOP2 [5]

33

34

Dynamic Multi-objective Optimization
Benchmarks

In the field of DMOO [5]:
- No standard set of benchmark functions
- Difficult to compare DMOO algorithms

Furthermore:
- Currently mostly only boundary-constrained DMOOPs
- Most real-world problems have constraints [7]
- Additional complexity being added:
  - Static objectives, dynamic constraints
  - Dynamic objectives, static constraints
  - Dynamic objectives, dynamic constraints

Dynamic Multi-objective Optimization
Benchmarks
Dealing with Constraints

- Three main approaches:
  - Penalty approaches
  - Adapt the Pareto-dominance relation to take into account constraint violation
  - Add constraint violation as an objective function to minimize
- Constraints can change over time

35

36

Dynamic Multi-objective Optimization
Benchmarks
Dealing with Constraints

Penalty approaches

- Add an additional term to individual's fitness calculation, i.e. constraint violation
- Death penalty – individuals with any constraint violation are rejected and no knowledge extracted from them
- Static penalty:
  - Penalty independent of current generation and remains static throughout run
  - Penalty is typically weighted sum of constraint violations
- Dynamic penalty – generation number is used in penalty calculations
- Adaptive penalty – information gathered from search process determines amount of penalty

Dynamic Multi-objective Optimization
Benchmarks
Dealing with Constraints

Adapting Pareto-dominance relation

Solution *i* is constrained dominant over a solution *j* if:

- Solution *i* is feasible and solution *j* is not
- Both solutions are infeasible, but solution *i* violates the constraints less than solution *j*
- Both solutions are feasible and *i* dominates *j* using Pareto-dominance

Dynamic Multi-objective Optimization
Benchmarks
Dealing with Constraints

Adding an objective function

- Minimize an additional objective function, which is the sum of constraint violations
- An additional objective
- If already optimizing 3 objectives, changes it into a many-objective problem

Analyzing Algorithms' Performance
Performance Measures

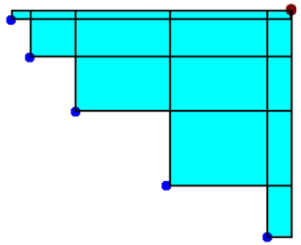
Dynamic Multi-objective Optimization
Performance Measures

In the field of DMOO:
- ❖ No standard set of performance measures
- ❖ Want to compare algorithms:
  - Typically just adapt static MOO measures
  - Typically report just averages and standard deviation
  - BUT, want to understand tracking ability and behaviour of algorithm

41

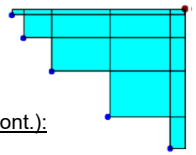
Dynamic Multi-objective Optimization
Performance Measures

Be careful when using measures from static MOO [9]:
- Often use hypervolume (Lebesque integral) to measure performance
- Reference vector is the worst values for each objective



42

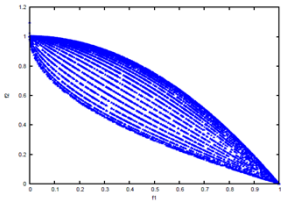
Dynamic Multi-objective Optimization
Performance Measures



Be careful when using measures from static MOO (cont.):
What happens if:
- An algorithm does not detect that a change occurred
- The new environment's POF results in a lower HV value than the previous environment's POF

HV of algorithm that got stuck will be better than the one that tracks the POF

HVR of algorithm that got stuck will be greater than one
⇒ indicates that we need to check whether algorithm got stuck



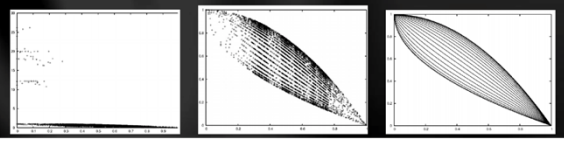
POF of dMOP2 [5]

43

Dynamic Multi-objective Optimization
Performance Measures

Be careful when using measures from static MOO (cont.):

When the environment changes frequently
- Algorithm finds non-dominated solutions further away from *POF*
- In time available, algorithm does not find any solutions that dominate outliers
  ⇒ *POF** for a specific time step may contain outliers
- Issue of outliers is applicable to both SMOO and DMOO



Outliers will skew distance-based measures such as GD, IGD, etc.

44

Dynamic Multi-objective Optimization
Performance Measures

Measures proposed by Camára et al [10]:

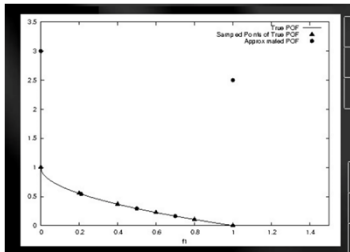
- $acc(t) = \frac{HV(POF^*(t))}{HV_{max}(POF^*(t))}$ Accuracy
- $acc_{alt}(t) = |HV(POF'(t)) - HV(POF^*(t))|$ Alternative accuracy
- $stab(t) = \max\{0, acc(t-1) - acc(t)\}$ stability
- $NS(t)$ is #non-dominated solutions found at time t

Also, reactivity

What do I learn if we report on the average and standard deviation values?

Dynamic Multi-objective Optimization
Performance Measures

Be careful when using measures from static MOO (cont.):

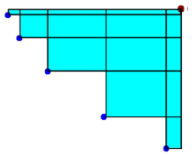


Outliers	GD	VD	MS
Yes	2.05565	4.596574	0.91833
No	0.00942	0.016311	0.4342

Outliers	HV	HVR	acc_{alt}
Yes	2.49898	0.84461	0.45974
No	0.69798	0.91994	0.06074

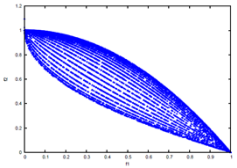
Analyzing Algorithms' Performance
Ensuring a Fair Comparison

Analyzing Algorithms' Performance
Ensuring Fair Comparison



In dynamic environments, how to you ensure a fair comparison?

- Give each algorithm a maximum #function evaluations per environment?
- Same #entities?
- And if archives are used by one and not another? Multiple archives?
- One using sentries and the other not?



POF of dMOP2 [5]

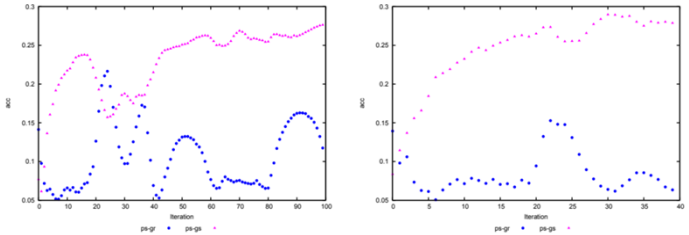
Analyzing Algorithms' Performance
Analyzing Results

49

Analyzing Algorithms' Performance
Analyzing Results

How can we solve this problem and measure the performance over time?

Can draw figures that show how measure changes over time…



That will be a lot of figures per study!

[4]

For each benchmark, different frequency and severity of change!

50

Analyzing Algorithms' Performance
Analyzing Results Wins-losses approach A (static) [10]

Per measure, per benchmark:

Algorithm 1	Algorithm 2	…	Algorithm n
Value1	Value1		Value1
Value2	Value2		Value2
…	…		…
Value30	Value30		Value30

Each value is avg PM value for environment

For SOOP: Each value is PM value for run

Kruskal-Wallis test

Statistical significant difference

For each pair of algorithms

Algorithm 1	Algorithm 2
Solution1	Solution1
Solution2	Solution2
…	…
Solution30	Solution30

Statistical significant difference

Avg of measure:

Alg1 > Alg2:
Alg1 #wins++
Alg2 #losses--

For each algorithm

Diff=#wins - #losses

Pair-wise Mann Whitney U

51

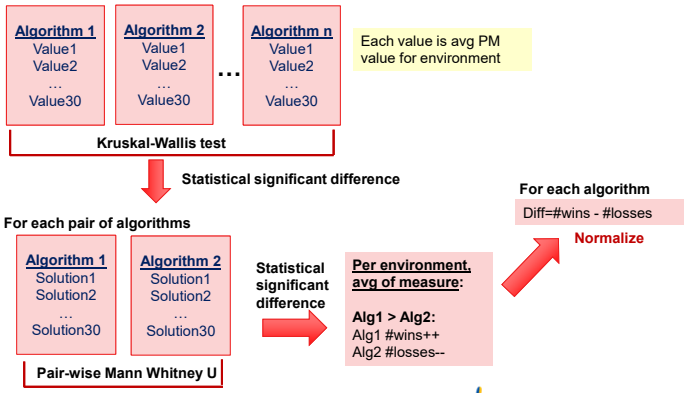
Analyzing Algorithms' Performance
Analyzing Results Wins-losses approach A (static) [10]

1. Calculate avg. performance measure value for each time step just before a change, over the number of runs, for each DMOA
2. Perform Kruskal-Wallis test on the performance measure values ⇒ Statistical significant difference between DMOA' performance?
3. If there is a statistical significant difference:
   - Perform pair-wise Mann-Whitney U tests ⇒ Statistical significant difference between the two DMOA' performance?
   - If there is a statistical significant difference:
     - Award a win to the DMOA with the best average performance measure value
     - Award a loss to other DMOA
4. Calculate for each DMOA the difference between its number of wins and number of losses ($Diff = \#wins - \#losses$)
5. Rank the DMOAs based on $Diff$.

52

Analyzing Algorithms' Performance
Analyzing Results — Wins-losses approach B (dynamic) [10]

Per measure, per benchmark:



53

Analyzing Algorithms' Performance
Analyzing Results — Wins-losses approach B (dynamic) [10]

1. Calculate avg. performance measure value for each time step just before a change, over the number of runs, for each DMOA
2. Perform Kruskal-Wallis test on the performance measure values ⇒ Statistical significant difference between DMOA' performance?
3. If there is a statistical significant difference:
   - Perform pair-wise Mann-Whitney U tests ⇒ Statistical significant difference between the two DMOA' performance?
   - If there is a statistical significant difference, for each environment:
     - Award a win to the DMOA with the best average performance measure value
     - Award a loss to other DMOA
4. Calculate for each DMOA the difference between its number of wins and number of losses ($Diff = \#wins - \#losses$)
5. Rank the DMOAs based on *Diff*. Normalize values for DMOOP

54

Analyzing Algorithms' Performance
Analyzing Results — Wins-losses approach B (dynamic) [10]

When solving DMOOPs, *Diff* can be calculated:

- per performance measure over all DMOOPs
- per environment type (for example per n_t-τ_t combination) over all DMOOPs
- over all performance measures and all DMOOPs
- per DMOOP type:
  - per performance measure over all DMOOPs of the specific DMOOP type
  - per environment type over all DMOOPs of the specific DMOOP type
  - over all performance measures and all DMOOPs of the specific DMOOP type

Can analyze results for various categories of DMOOPs

55

Analyzing Algorithms' Performance
Analyzing Results

Overall results with regards to performance measures:

PM	Results	DMOO Algorithm				
		DNSGA-II-A	DNSGA-II-B	dCOEA	DMOPSO	DVEPSO
acc	Wins	94	109	129	118	119
acc	Losses	120	104	164	86	95
acc	Diff	-26	5	-35	32	24
acc	Rank	4	3	5	**1**	2
stab	Wins	67	94	39	117	96
stab	Losses	89	66	200	39	50
stab	Diff	-22	28	-161	78	46
stab	Rank	4	3	5	**1**	2
NS	Wins	185	187	116	53	111
NS	Losses	83	78	202	195	129
NS	Diff	102	109	-86	-142	-18
NS	Rank	2	**1**	4	5	3

More wins than losses (more statistically significant better results)
More losses than wins (more statistically significant worse results)

56

Dynamic Multi-objective Optimization
Analyzing Results

n_t	τ_t	PM	Results	DMOO Algorithm				
				DNSGA-II-A	DNSGA-II-B	dCOEA	DMOPSO	DVEPSO
10	10	all	Wins	35	33	22	56	44
10	10	all	Losses	36	35	76	16	27
10	10	all	Diff	-1	-2	-54	40	17
10	10	all	Rank	3	4	5	**1**	2
10	25	all	Wins	27	30	23	43	29
10	25	all	Losses	28	26	56	14	28
10	25	all	Diff	-1	4	-33	29	1
10	25	all	Rank	4	2	5	**1**	3
10	50	all	Wins	22	27	27	33	21
10	50	all	Losses	25	22	44	17	22
10	50	all	Diff	-3	5	-17	16	-1
10	50	all	Rank	4	2	5	**1**	3
1	10	all	Wins	30	34	44	41	43
1	10	all	Losses	50	40	46	29	27
1	10	all	Diff	-20	-6	-2	12	16
1	10	all	Rank	5	4	3	2	**1**
20	10	all	Wins	24	25	56	44	32
20	10	all	Losses	42	39	50	19	31
20	10	all	Diff	-18	-14	6	25	1
20	10	all	Rank	5	4	2	**1**	3

More wins than losses (more statistically significant better results)

More losses than wins (more statistically significant worse results)

Real-world Applications

Real-world Applications

Real-world Application Areas:

- Costing optimization, e.g. electricity
- Design optimization
- Solving inverse kinematics (robotics, character animation)
- Financial markets – currency exchange/portfolio optimization [11]
- Clustering of dynamic data [12]

Real-world Applications

Real-world Application Areas (cont.):

- Dynamic scheduling:
  - Network [13]
  - Smart home appliances [14]
  - Hydro-thermal power / Greenhouse control [15]
  - Railway junction [16]
  - Raw ore allocation in mineral processing [17]
  - Route planning [18]

Challenges in DMOO

61

Challenges in DMOO

- Decision Making
- Algorithm Behaviour

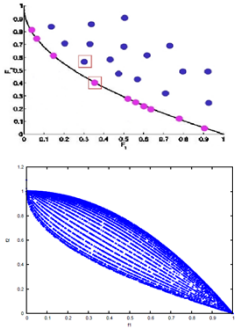
62

Challenges in DMOO
Decision Making

- Show POF to decision maker and s/he selects a solution, or
- Allow decision maker to guide the search process by:
  - Incorporating their preferences into search
  - Limiting search to a certain area of POF

Best way of doing this?

Dynamic – after a change?



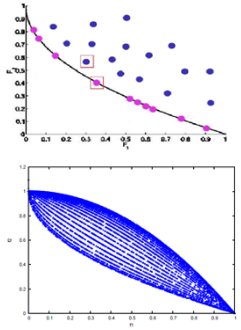
POF of dMOP2 [5]

63

Challenges in DMOO
Decision Making

Show POF to decision maker and s/he selects a solution:

- May find an even more optimal weight combination
- Find different possibilities where each solution is an "optimal weight combination"

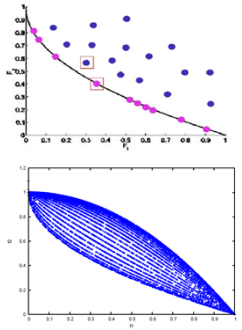


POF of dMOP2 [5]

64

Challenges in DMOO
Decision Making

- Not much research here
- Focus more on the algorithms
- How do we assist the decision maker?



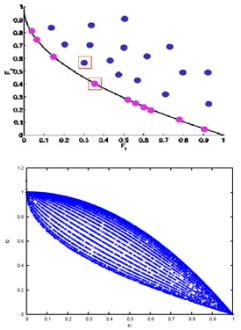
POF of dMOP2 [5]

65

Challenges in DMOO
Decision Making

- Can allow decision maker to indicate preference by selecting a region on the POF
- How do we use/deal with this preference in new environments?

Preference-based optimization



POF of dMOP2 [5]

66

Challenges in DMOO
Algorithm Behavior

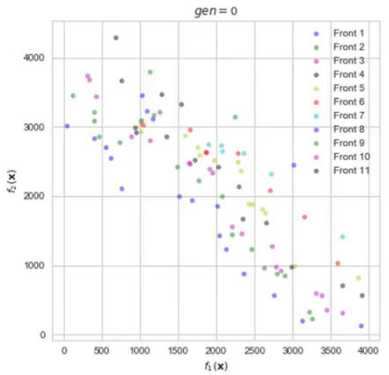
- Understanding search behavior of algorithm over time – how do changes affect it?
- Understanding when diverging, converging
- Understanding behavior of entities over entire run
- Visualise this in an understandable manner

→ Best way of doing this?

→ Dynamic – after a change?

67

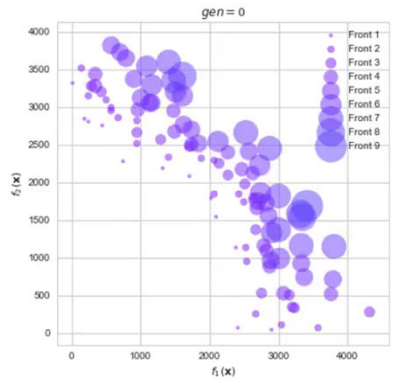
Challenges in DMOO
Algorithm Behavior



[20]

68

Challenges in DMOO
Algorithm Behavior



[20]

69

Emerging Areas

70

Emerging Areas

- Many-objective Optimization
- Fitness Landscape Analysis
- Solving Real-world Problems

71

Emerging Areas
Many-objective Optimization

- More than 3 objectives → many-objective optimization
- Pareto-dominance principle cannot be used to guide search
- As #objectives increase, #non-dominated solutions increase
- Require new approaches to guide the search process

What if we have dynamic many-objective optimization?

72

Emerging Areas
Fitness Landscape Analysis

- Why does an algorithm perform well on certain types of problems and not on others?
- When trying to solve a problem, which algorithm should I choose?
- Need to categorise the problems, but how?

What if we have dynamic search spaces?

Emerging Areas
Solving Real-world Problems

- Currently algorithms are mostly tested on either benchmark problems or real-world problems
  - ➡ No feedback loop, no connection
- Issues identified when solving real-world problems should feed back into the research challenges of the field and benchmark problems

Conclusions

Conclusions

- Many real-world problems are dynamic in nature
- Dynamic MOO (DMOO) has exciting challenges/opportunities
- Need to apply the algorithms to more real-world problems
- Feed the learning back into the field
- Emerging areas

Join Us!

- Many research opportunities!
- Do you want to join the DMOO community?
- Do you want to collaborate?
- E-mail me: m.helbig@griffith.edu.au

77

References

78

References

[1] Pixabay License. Available online at: https://pixabay.com/vectors/search/blue%20car%20car/

[2] Guillaume Jackquenot. CC BY-SA 3.0, via Wikimedia Commons. Available online at: https://commons.wikimedia.org/wiki/File:NonConvex.gif#globalusage

[3] Design Impact. Introduction to scalarization methods for multi-objective optimization. Available online at: https://www.youtube.com/watch?v=yc9NwvlpEpI&t=2089s

[4] M. Helbig. Solving dynamic multi-objective optimization problems using vector evaluated particle swarm optimization, PhD thesis, University of Pretoria, 2012. Available online at: https://repository.up.ac.za/handle/2263/28161

[5] M. Helbig and A.P. Engelbrecht. Benchmarks for dynamic multi-objective optimisation algorithms. ACM Computing Surveys, 46(3), 2014. Available online at: https://dl.acm.org/doi/10.1145/2517649

[6] Petr Krtochvil. CCO Public Domain License. Available online at: https://www.publicdomainpictures.net/en/view-image.php?image=38393&picture=meerkat-on-watch

[7] M. Helbig. Challenges of applying dynamic multi-objective optimization to real-world problems. *Women in Computational Intelligence: Key Advances and Perspectives on Emerging Topics*, Women in Engineering and Science, Springer. To be published.

79

References (cont.)

[8] M. Helbig and A.P. Engelbrecht. Population-based metaheuristics for continuous boundary-constrained dynamic multi-objective optimisation problems. *Swarm and Evolutionary Computation*, 14:31-47, 2014. Available online at: https://www.sciencedirect.com/science/article/abs/pii/S2210650213000539

[9] M. Helbig and A.P. Engelbrecht. Issues with performance measures for dynamic multi-objective optimization. *In Proceedings of the IEEE Symposium Series on Computational Intelligence*, Singapore, 16-19 April, 2013. Available online at: https://ieeexplore.ieee.org/document/6595767

[10] M. Helbig and A.P. Engelbrecht. Analysing the performance of dynamic multi-objective optimization algorithms. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, Cancun, Mexico, 20-23 June, 2013. Available online at: https://ieeexplore.ieee.org/document/6557744

[11] F. Atiah and M. Helbig. Effects of decision models on dynamic multi-objective optimization algorithms for financial markets. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, pp. 762–770, Wellington, New Zealand, 10-13 June, 2019. Available online at: https://ieeexplore.ieee.org/document/8790275

80

References (cont.)

[12] K. Kim, R. McKay and B. R. Moon. Multiobjective evolutionary algorithms for dynamic social network clustering. In *Proceedings of the Annual Conference on Genetic and Evolutionary Computation*, pp. 1179–1186. Portland, Oregan, USA, 2010. Available online at: https://dl.acm.org/doi/10.1145/1830483.1830699

[13] G. Ismayilov and H. Topcuoglu. Neural network based multi-objective evolutionary algorithm for dynamic workflow scheduling in cloud computing. *Future Generation Computer Systems*, 102:307–322, 2020. Available online at: https://www.sciencedirect.com/science/article/abs/pii/S0167739X19306983

[14] W. Trabelsi, et al. Leveraging evolutionary algorithms for dynamic multi-objective optimization scheduling of multi-tenant smart home appliances. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), pp. 3533–3540. Vancouver, Canada, 2016. Available online at: https://ieeexplore.ieee.org/document/7744237

[15] K. Deb, N. Rao and S. Karthik. Dynamic multi-objective optimization and decision-making using modified NSGA-II. In *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, vol. 4403, pp. 803–817. Springer, Berlin, Heidelberg, 2007. Available online at: https://link.springer.com/chapter/10.1007/978-3-540-70928-2_60

81

References (cont.)

[16] J. Eaton, S. Yang and M. Gongora. Ant colony optimization for simulated dynamic multiobjective railway junction rescheduling. *IEEE Transactions on Intelligent Transportation Systems*, 18(11), 2980–2992, 2017. Available online at: https://ieeexplore.ieee.org/document/7875408

[17] J. Ding, et al. Dynamic evolutionary multiobjective optimization for raw ore allocation in mineral processing. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 3(1), 36–48, 2019. Available online at: https://ieeexplore.ieee.org/document/8331273

[18] H.K.K. Hara. Hybrid genetic algorithm for dynamic multi-objective route planning with predicted traffic in a real-world road network. In *Proceedings of the Annual Conference on Genetic and Evolutionary Computation (GECCO)*, pp. 657–664. Atlanta, Georgia, USA, 2008. Available online at: https://dl.acm.org/doi/10.1145/1389095.1389226

[19] Nations Online Project. Map of Australia. Available online at: https://www.nationsonline.org/oneworld/map/australia-map.htm

[20] M. Helbig. Visualising the search process for multi-objective optimization. In Proceedings of the Genetic and Evolutionary Computation Conference Companion, p. 1560-1561, Kyoto, Japan, 15-19 July, 2018. Available online at: https://dl.acm.org/doi/10.1145/3205651.3208314

82

83