合肥学院
HEFEI UNIVERSITY

Metaheuristics for Smart Manufacturing

8. Outlook

Thomas Weise · 汤卫思

tweise@hfuu.edu.cn · http://iao.hfuu.edu.cn

Hefei University, South Campus 2 | 合肥学院 南艳湖校区/南2区
Faculty of Computer Science and Technology | 计算机科学与技术系
Institute of Applied Optimization | 应用优化研究所
230601 Shushan District, Hefei, Anhui, China | 中国 安徽省 合肥市 蜀山区 230601
Econ. & Tech. Devel. Zone, Jinxiu Dadao 99 | 经济技术开发区 锦绣大道99号

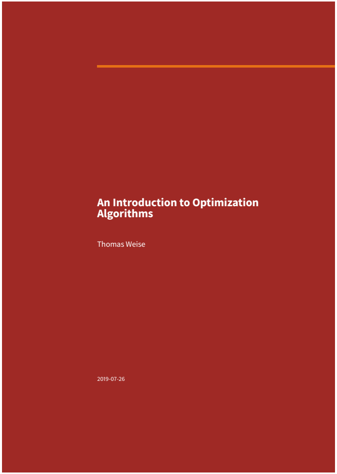
1. What else can we do?
2. Limits
3. Summary

The slides are available at http://iao.hfuu.edu.cn/155, the book at http://thomasweise.github.io/aitoa, and the source code at http://www.github.com/thomasWeise/aitoa-code

course book

course material

The contents of this course are available as free electronic book *"An Introduction to Optimization Algorithms"* [1] at http://thomasweise.github.io/aitoa in pdf, html, azw3, and epub format, created with our bookbuildeR tool chain.



Section Outline

- Advanced metaheuristics like SA or population-based methods like Evolutionary Algorithms perform global search and can obtain better results than simple local searches like hill climbers.

Advanced Metaheuristics

- Advanced metaheuristics like SA or population-based methods like Evolutionary Algorithms perform global search and can obtain better results than simple local searches like hill climbers.
- In our experiments, we did not try to squeeze out the last bit of performance out of each algorithm.

- Advanced metaheuristics like SA or population-based methods like Evolutionary Algorithms perform global search and can obtain better results than simple local searches like hill climbers.
- In our experiments, we did not try to squeeze out the last bit of performance out of each algorithm.
- We focussed on simple, understandable concepts.

- Advanced metaheuristics like SA or population-based methods like Evolutionary Algorithms perform global search and can obtain better results than simple local searches like hill climbers.
- In our experiments, we did not try to squeeze out the last bit of performance out of each algorithm.
- We focussed on simple, understandable concepts.
- By now, you have a general idea about the basic principles of metaheuristics and how they can be applied to problems from smart manufacturing.

- Advanced metaheuristics like SA or population-based methods like Evolutionary Algorithms perform global search and can obtain better results than simple local searches like hill climbers.
- In our experiments, we did not try to squeeze out the last bit of performance out of each algorithm.
- We focussed on simple, understandable concepts.
- By now, you have a general idea about the basic principles of metaheuristics and how they can be applied to problems from smart manufacturing.
- They are easy to use and fast to implement.

- Advanced metaheuristics like SA or population-based methods like Evolutionary Algorithms perform global search and can obtain better results than simple local searches like hill climbers.
- In our experiments, we did not try to squeeze out the last bit of performance out of each algorithm.
- We focussed on simple, understandable concepts.
- By now, you have a general idea about the basic principles of metaheuristics and how they can be applied to problems from smart manufacturing.
- They are easy to use and fast to implement.
- But there is so much more, much more can be done!

What else can be done?

- We could have tried using a different representation.

What else can be done?

- We could have tried using a different representation.
- Picking a good representation where search operators can make progress is very very important in metaheuristic optimization! [2, 3].

What else can be done?

- We could have tried using a different representation.
- Picking a good representation where search operators can make progress is very very important in metaheuristic optimization! [2, 3].
- Global search yields good results but is slow

What else can be done?

- We could have tried using a different representation.
- Picking a good representation where search operators can make progress is very very important in metaheuristic optimization! [2, 3].
- Global search yields good results but is slow while local search is fast but prone to premature convergence

What else can be done?

- We could have tried using a different representation.
- Picking a good representation where search operators can make progress is very very important in metaheuristic optimization! [2, 3].
- Global search yields good results but is slow while local search is fast but prone to premature convergence – why not combine the two?

What else can be done?

- We could have tried using a different representation.
- Picking a good representation where search operators can make progress is very very important in metaheuristic optimization! [2, 3].
- Global search yields good results but is slow while local search is fast but prone to premature convergence – why not combine the two?
- The resulting "Memetic Algorithms" [4, 5] often yield the best results! [6]

What else can be done?

- We could have tried using a different representation.
- Picking a good representation where search operators can make progress is very very important in metaheuristic optimization! [2, 3].
- Global search yields good results but is slow while local search is fast but prone to premature convergence – why not combine the two?
- The resulting "Memetic Algorithms" [4, 5] often yield the best results! [6]
- But these are other stories. . .

Section Outline

1. What else can we do?
2. Limits
3. Summary

Scalability

- Time required for solving $\mathcal{NP}$-hard problems grows exponential with input size

Scalability

- Time required for solving $\mathcal{NP}$-hard problems grows exponential with input size
- Metaheuristic optimization: approximately solve $\mathcal{NP}$-hard problems in feasible time

Scalability

- Time required for solving $\mathcal{NP}$-hard problems grows exponential with input size
- Metaheuristic optimization: approximately solve $\mathcal{NP}$-hard problems in feasible time
- ...but their time requirement also grows with problem size...

Scalability

- Time required for solving $\mathcal{NP}$-hard problems grows exponential with input size
- Metaheuristic optimization: approximately solve $\mathcal{NP}$-hard problems in feasible time
- …but their time requirement also grows with problem size…
- "Curse of Dimensionality": solution space volume increases exponentially with number of decision variables (genes) [7, 8]

Scalability

- Time required for solving $\mathcal{NP}$-hard problems grows exponential with input size
- Metaheuristic optimization: approximately solve $\mathcal{NP}$-hard problems in feasible time
- ... but their time requirement also grows with problem size...
- "Curse of Dimensionality": solution space volume increases exponentially with number of decision variables (genes) [7, 8]
- Example: search in $(1 \ldots 10)^n$

Scalability

- Time required for solving $\mathcal{NP}$-hard problems grows exponential with input size
- Metaheuristic optimization: approximately solve $\mathcal{NP}$-hard problems in feasible time
- ...but their time requirement also grows with problem size...
- "Curse of Dimensionality": solution space volume increases exponentially with number of decision variables (genes) [7, 8]
- Example: search in $(1 \ldots 10)^n$
- any algorithm (for non-trivial problems) takes longer for larger inputs...

Scalability

- Time required for solving $\mathcal{NP}$-hard problems grows exponential with input size
- Metaheuristic optimization: approximately solve $\mathcal{NP}$-hard problems in feasible time
- …but their time requirement also grows with problem size…
- "Curse of Dimensionality": solution space volume increases exponentially with number of decision variables (genes) [7, 8]
- Example: search in $(1 \ldots 10)^n$
- any algorithm (for non-trivial problems) takes longer for larger inputs…
- This is one of the biggest obstacles for practical applications of metaheuristic optimization, in my opinion.

Law of Diminishing Returns

- The Law of Diminishing Returns [9] holds in two ways

- The Law of Diminishing Returns [9] holds in two ways:
  1. When I apply an optimization algorithm to an optimization problem, it will make most of the improvements in solution quality relatively early on.

- The Law of Diminishing Returns [9] holds in two ways:
  1. When I apply an optimization algorithm to an optimization problem, it will make most of the improvements in solution quality relatively early on. For more improvements, I will eventually need to wait longer and longer.

Law of Diminishing Returns

- The Law of Diminishing Returns [9] holds in two ways:
  1. When I apply an optimization algorithm to an optimization problem, it will make most of the improvements in solution quality relatively early on. For more improvements, I will eventually need to wait longer and longer. This makes sense: The closer I get to the real absolutely best, globally optimal, solution, the fewer solutions exist that are better than the one I have.

Law of Diminishing Returns

- The Law of Diminishing Returns [9] holds in two ways:
  1. When I apply an optimization algorithm to an optimization problem, it will make most of the improvements in solution quality relatively early on. For more improvements, I will eventually need to wait longer and longer. This makes sense: The closer I get to the real absolutely best, globally optimal, solution, the fewer solutions exist that are better than the one I have. Hence it takes longer and longer to find them.

- The Law of Diminishing Returns [9] holds in two ways:
  1. When I apply an optimization algorithm to an optimization problem, it will make most of the improvements in solution quality relatively early on. For more improvements, I will eventually need to wait longer and longer. This makes sense: The closer I get to the real absolutely best, globally optimal, solution, the fewer solutions exist that are better than the one I have. Hence it takes longer and longer to find them. If I found the global optimum, it will take infinitely long to improve further...

Law of Diminishing Returns

- The Law of Diminishing Returns [9] holds in two ways:
  1. When I apply an optimization algorithm to an optimization problem, it will make most of the improvements in solution quality relatively early on.
  2. I can develop better and better optimization algorithms.

Law of Diminishing Returns

- The Law of Diminishing Returns [9] holds in two ways:
  1. When I apply an optimization algorithm to an optimization problem, it will make most of the improvements in solution quality relatively early on.
  2. I can develop better and better optimization algorithms. But I need to invest more and more research and development time to make smaller and smaller improvements.

Law of Diminishing Returns

- The Law of Diminishing Returns [9] holds in two ways:
  1. When I apply an optimization algorithm to an optimization problem, it will make most of the improvements in solution quality relatively early on.
  2. I can develop better and better optimization algorithms. But I need to invest more and more research and development time to make smaller and smaller improvements. We have the biggest improvements from single random sample to may random samples and from there to the simple Hill Climber. From then to HC+restarts and from there to EAs, the improvements were smaller. (SA is another improvement of HC, not the EA.)

Law of Diminishing Returns

- The Law of Diminishing Returns [9] holds in two ways:
  1. When I apply an optimization algorithm to an optimization problem, it will make most of the improvements in solution quality relatively early on.
  2. I can develop better and better optimization algorithms. But I need to invest more and more research and development time to make smaller and smaller improvements. We have the biggest improvements from single random sample to may random samples and from there to the simple Hill Climber. From then to HC+restarts and from there to EAs, the improvements were smaller. (SA is another improvement of HC, not the EA.)
- But in an enterprise, even 1% of improvement can mean millions of RMB...

Summary

- You now know the basic concepts behind metaheuristic optimization.

- You now know the basic concepts behind metaheuristic optimization.
- You know how to take an optimization problem, maybe from the real world, and then how to develop, implement, and evaluate and optimization approach for it.

Summary

- You now know the basic concepts behind metaheuristic optimization.
- You know how to take an optimization problem, maybe from the real world, and then how to develop, implement, and evaluate and optimization approach for it.
- The ideas we have exercised here can be applied in many domains.

Summary

- You now know the basic concepts behind metaheuristic optimization.
- You know how to take an optimization problem, maybe from the real world, and then how to develop, implement, and evaluate and optimization approach for it.
- The ideas we have exercised here can be applied in many domains.
- You also know at least some of the limiting factors in our field.

Summary

- You now know the basic concepts behind metaheuristic optimization.
- You know how to take an optimization problem, maybe from the real world, and then how to develop, implement, and evaluate and optimization approach for it.
- The ideas we have exercised here can be applied in many domains.
- You also know at least some of the limiting factors in our field.
- I hope this course was useful and interesting to you.

Summary

- You now know the basic concepts behind metaheuristic optimization.
- You know how to take an optimization problem, maybe from the real world, and then how to develop, implement, and evaluate and optimization approach for it.
- The ideas we have exercised here can be applied in many domains.
- You also know at least some of the limiting factors in our field.
- I hope this course was useful and interesting to you.
- You can find a much more comprehensive discussion in our book [1].

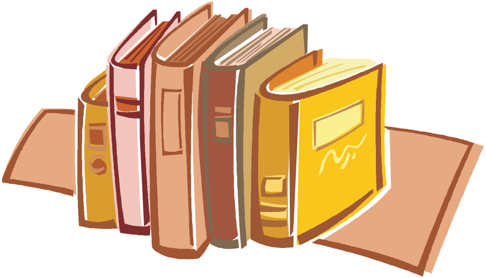
谢谢
Thank you

Thomas Weise [汤卫思]
tweise@hfuu.edu.cn
http://iao.hfuu.edu.cn

Hefei University, South Campus 2
Institute of Applied Optimization
Shushan District, Hefei, Anhui, China

Caspar David Friedrich, “Der Wanderer über dem Nebelmeer”, 1818
http://en.wikipedia.org/wiki/Wanderer_above_the_Sea_of_Fog

Bibliography



Bibliography I

1. Thomas Weise. *An Introduction to Optimization Algorithms*. Institute of Applied Optimization (IAO), Faculty of Computer Science and Technology, Hefei University, Hefei, Anhui, China, 2019-06-25 edition, 2018–2019. URL `http://thomasweise.github.io/aitoa/`. see also [3].
2. Franz Rothlauf. *Representations for Genetic and Evolutionary Algorithms*, volume 104 of *Studies in Fuzziness and Soft Computing book series (STUDFUZZ)*. Springer-Verlag, Berlin/Heidelberg, 2 edition, 2006. doi: 10.1007/3-540-32444-5.
3. Thomas Weise. *Global Optimization Algorithms – Theory and Application*. it-weise.de (self-published), Germany, 2009. URL `http://www.it-weise.de/projects/book.pdf`.
4. Pablo Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech Concurrent Computation Program C3P 826, California Institute of Technology (Caltech), Caltech Concurrent Computation Program (C3P), Pasadena, CA, USA, 1989. URL `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.27.9474`.
5. William Eugene Hart, James E. Smith, and Natalio Krasnogor, editors. *Recent Advances in Memetic Algorithms*, volume 166 of *Studies in Fuzziness and Soft Computing book series (STUDFUZZ)*. Springer, Berlin, Heidelberg, 2005. ISBN 978-3-540-22904-9. doi: 10.1007/3-540-32363-5.
6. Thomas Weise, Raymond Chiong, Ke Tang, Jörg Lässig, Shigeyoshi Tsutsui, Wenxiang Chen, Zbigniew Michalewicz, and Xin Yao. Benchmarking optimization algorithms: An open source framework for the traveling salesman problem. *IEEE Computational Intelligence Magazine (CIM)*, 9:40–52, August 2014. doi: 10.1109/MCI.2014.2326101.
7. Richard Ernest Bellman. *Dynamic Programming*. Dover Books on Mathematics. Princeton University Press, Princeton, NJ, USA, 1957. ISBN 0486428095.
8. Richard Ernest Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton Legacy Library. Princeton University Press, Princeton, NJ, USA, 1961. ISBN 0691079013.
9. Paul Anthony Samuelson and William Dawbney Nordhaus. *Microeconomics*. McGraw-Hill Education (ISE Editions), Boston, MA, USA, 17 edition, 2001. ISBN 0071-180664.