Fitness Landscapes and Problem Difficulty

Jean-Paul Watson
Sandia National Laboratories
Albuquerque, New Mexico
jwatson@sandia.gov

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

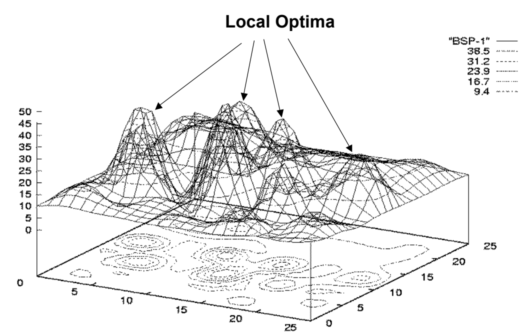
Talk Outline

- What is a fitness landscape?
- Why should algorithm designers care about the fitness landscape?
- How do you tell if a fitness landscape feature matters?
  - Instance versus ensemble-level problem difficulty
  - How important are "well-known" landscape features?
- Linking fitness landscape structure and algorithm run-time dynamics
  - An illustrative example from Job-Shop Scheduling
- Future research areas in fitness landscape analysis
- Conclusions

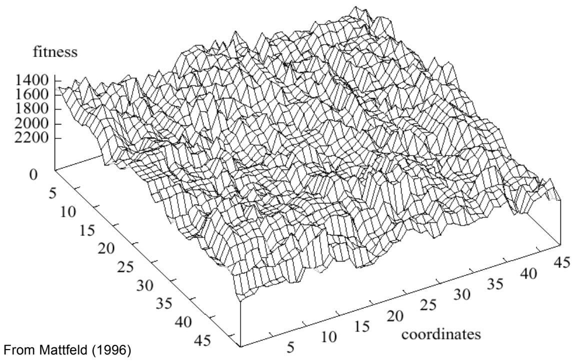
What is a Fitness Landscape?

- For typical local search methods (tabu search, simulated annealing)
  - A vertex-weighted graph!
  - Three core components
    - A search space *S*
    - A fitness or objective function f:*S*->*R*
    - A move operator *N*:*S*->*P(S)*
  - <u>*To a first-order approximation*</u> – see Reeves (1998) for critique
- For evolutionary algorithms
  - The picture is significantly less clear
  - Multiple move operators
  - Move operators that take multiple solutions (e.g., crossover)
  - See Jones (1995) for a great discussion of these and other related issues

Local Search and the Fitness Landscape



A (Slightly) More Realistic Example



Taken From Mattfeld (1996)

More Complexities and Subtleties

- Two qualitatively different types of fitness landscape
- "Type 1" Fitness Landscapes
  - Dominated by large plateaus of equally fit solutions
  - Different terminology (e.g., benches and exits)
  - Not hard to find in combinatorial optimization
    - E.g., MAX-SAT and flow-Shop Scheduling
- "Type 2" Fitness Landscapes
  - Dominated by local optima, distinct neighbor fitness values
  - Different terminology (e.g., barriers and depth)
  - Pervasive in function/global optimization
  - The "other half" of combinatorial optimization problems
    - E.g., the TSP
- See Hoos and Stutzle (2005) for further information

Why Should You Care About Fitness Landscapes?

- *The* motivating observation
  - Algorithm performance depends on the ability of a search strategy to exploit the structure of the underlying fitness

- Implications
  1. Knowledge of fitness landscape structure is the *only* way to design algorithms in a *targeted* fashion, i.e., not hacking
  2. Algorithms are necessarily "tuned" to a particular class of fitness landscapes => you have to know your problem!

- Caveat
  - Fitness landscape structure is important, but cannot in truth be studied independently of the algorithm under consideration
  - *Algorithm behavior and fitness landscape structure are linked*

Fitness Landscape Features: An Overview (1)

- Correlation length
  - Weinberger, Stadler
  - Generate a fitness time-series via a random walk
  - Autocorrelation measures ruggedness
  - Rugged landscapes => more difficulty for adaptive algorithms

- Fitness-distance correlation
  - Kirkpatrick and Toulouse, Boese et al, Jones and Forrest
  - Generate a large sample of random local optima
  - Compute the correlation between
    - Distance-to-best or average distance to other optima
    - Fitness
  - Strong correlation => good solutions are clustered
    - The "big-valley" structure
  - Weak correlation => adaptive search will lead you astray

Fitness Landscape Features: An Overview (2)

- Barrier structure
  - The entire simulated annealing research community!
  - How much of a fitness decrease is required to escape the attractor basin of a local optimum?
  - Barrier trees (Stadler)
    - Is search likely to be trapped in certain regions of the search space?
    - Leonard-Jones clusters
- The average distance between local optima
  - Mattfeld
  - What is the average distance between local optima?
  - Quantifies search space "diameter" or "width"
  - Large search spaces => higher degree of difficulty

Fitness Landscape Features: An Overview (3)

- The number of optimal solutions
  - Clark et al.
  - How many *globally* optimal solutions are there?
  - More optimal solutions => they should be easier to find
  - Popularized in the context of MAX-SAT
- Backbone size
  - Slaney and Walsh, Singer et al.
  - How many solution attributes are found in *all* optimal solutions?
  - Large backbone => once you "solve" the backbone, the rest of the problem should be easy
- The average distance between local optima and optimal solutions
  - Singer et al.
  - What is the average distance between local optima and the *nearest* optimal solution?
  - Simultaneously accounts for both search space size and the number of "targets" embedded within the sub-space

How To Tell If a Fitness Landscape Feature "Matters"?

- Intuition
  - A fitness landscape feature is important if its presence is highly correlated with the difficulty of locating an optimal solution
  - In other words, if the presence of the feature impedes an search algorithm from operating effectively
- Some things to consider before undertaking analysis
- Do you care about ensemble-level differences in problem difficulty?
  - E.g., 30-city TSPs versus 100-city TSPs
- Do you care about instance-level differences in problem difficulty?
  - E.g., 1000 instances of 100-city TSPs
- An observation
  - Cost to solve 100-city TSPs varies over 8 *orders of magnitude*
- An opinion
  - If you can't account for such huge differences at the instance level, you can't hope to explain differences at the ensemble level

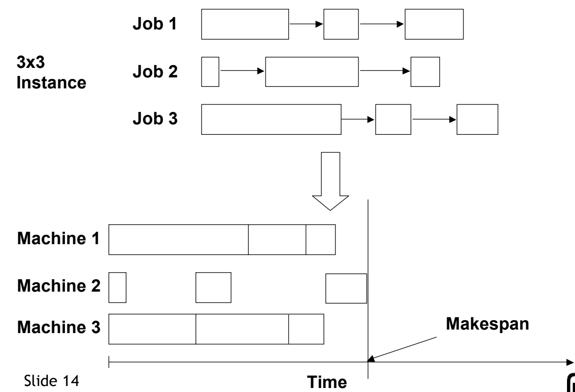
Static Cost Models of Problem Difficulty

- A static cost model
  - Accounts for the variability in problem difficulty observed in a set of fixed-dimension problem instances
- The "static" modifier derives from the fact that algorithm dynamics are not explicitly taken into account
- Problem difficulty
  - How much does it cost on average to locate an optimal solution to a given problem instance?
- Fixed-dimension problem instances
  - E.g., a set of 100 random Euclidean TSP instances
- Linear regression of landscape feature versus problem difficulty
- The r^2 value of the resulting model quantifies the proportion of variability in problem difficulty accounted for by the model

Static Cost Models: The Current Situation

- *Most well-known search space features are only weakly correlated with problem difficulty*
  - Correlation length
  - The number of optimal solutions
  - The average distance between local optima
  - The backbone size
  - Fitness-distance correlation
- These features *at best* account for 25%-50% of the total variability in problem difficulty on *small* problems
  - And often much less
- Accuracy rapidly drops as problem size increases

Experimental Domain: Job-Shop Scheduling (JSP)

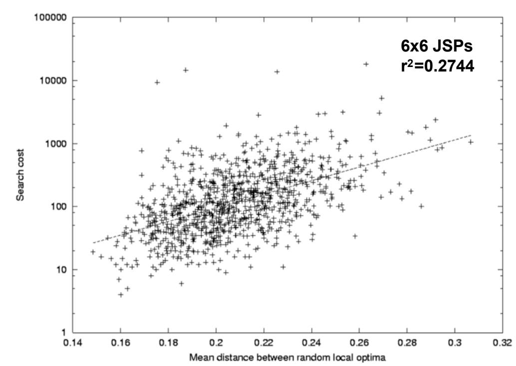


Performance of Static Cost Models on the JSP

- Consider a set of 1000 6-job, 6-machine instances
  - Small in comparison to any “real” benchmark problems
- Static cost model accuracy for widely studied measures
  - Correlation length $r^2=0.0$
  - The number of globally optimal solutions $r^2=0.2223$
  - The backbone size $r^2=0.2231$
  - Average distance between local optima $r^2=0.2744$
  - Fitness-distance correlation $r^2=0.1211$
- Only account for about 25% of the total variability
  - Why are these popular and widely studied?

- *Things get worse for larger problems*, e.g., 10-jobs, 10-machines

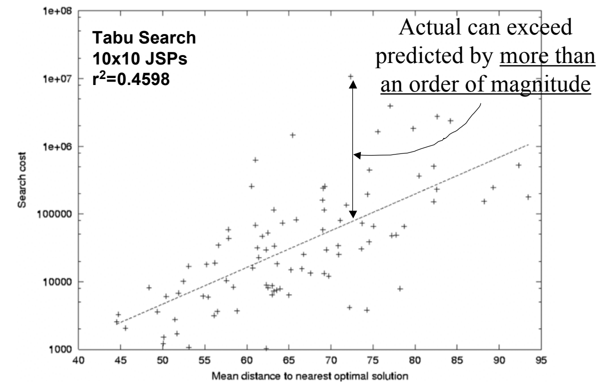
The Best of the Lot…



A More Effective Static Cost Model (1)

- Hypothesis:
  - Problem difficulty is proportional to the *effective* size of the search space
- Must simultaneously account for both
  - The absolute size of the search space
  - The number and distribution of solutions within the search space
- New /unexplored measure: $\overline{d}_{lopt-opt}$
  - The mean distance between random local optima and the nearest optimal solution

A More Effective Static Cost Model (2)



$\overline{d}_{lopt-opt}$

Static Cost Models and Landscape Features: Discussion

- It is not enough to simply posit that a specific fitness landscape feature plays an important role in problem difficulty
  - Intuition suggests that a particular feature "should" be important
  - Intuition is often wrong than right in science
- It is easy enough to subject these hypotheses to rigorous testing
  - Static cost models via linear regression
- A common theme
  - Features that are "thought" to be important for many widely-used algorithms aren't all that important at all
- Implication
  - Landscape analysis is not a "solved" research area

Beyond Static Cost Models: The Test Subjects

- Tabu search
  - Steepest-descent local search, but…
  - … prevents search from "undoing" recent moves
- Metropolis sampling (aka MCMC)
  - Always accept improving/equal moves
  - Probabilistically accept worse moves
- Iterated local search
  - Generate large "kick-moves" to escape local optima
  - Apply greedy descent and iterate…

Modeling Objectives

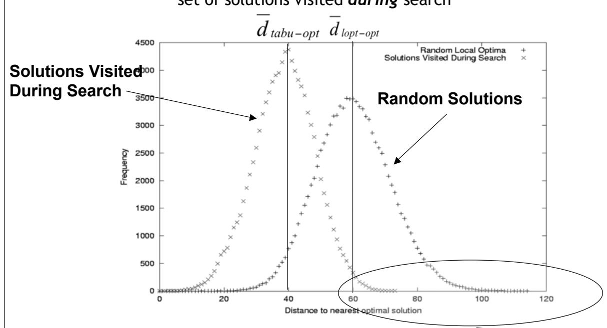
- To account for variability in problem difficulty
  - Difficulty = cost to locate an optimal solution
  - *Cost models* of local search algorithms
- To characterize the relationship between search space structure and problem difficulty
  - What features cause problems for local search?
- To model the run-time behavior of local search algorithms
  - What is the high-level search strategy?

Static Cost Models for the JSP: Summary

- New measure accounts for 65%-90% of the variability in problem difficulty for small JSPs…
- … but only 40-45% of the variability in large JSPs
- Conclusion
  - Problem difficulty is proportional to the effective size of the search space
  - But only to a first-order approximation
- Universal drawbacks to static cost models
  - Accuracy fails to scale to larger JSPs
  - No insight into run-time dynamics

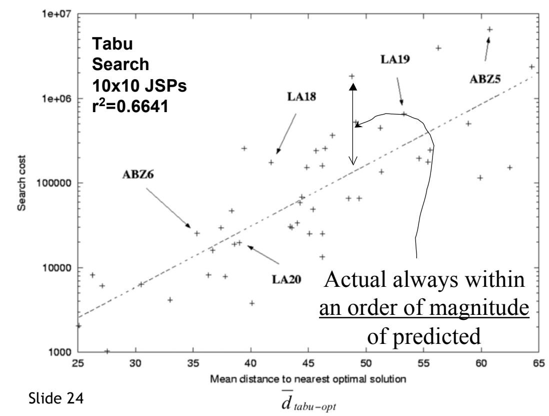
Bias and Tabu Search in the JSP

Observation: Random local optima are ***not*** necessarily representative of the set of solutions visited ***during*** search



Large differences in maximal distance!

Accuracy of the Quasi-Dynamic Model

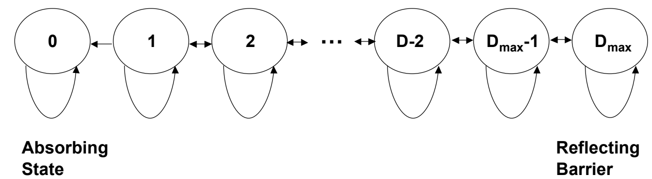


Dynamic Cost Models

- Any local search algorithm can *in principle* be modeled as a Markov chain
  - Finite number of states
  - Exact transition probabilities
- Is this approach tractable?
  - No!
- Key issues in developing tractable Markov models
  - How to aggregate solutions?
  - How to model short-term memory? (if applicable)

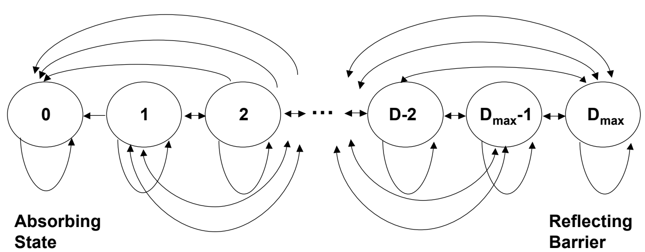
A Markov Model of Metropolis Sampling

- Aggregate solutions based on their distance to the nearest optimal solution
- A simple one-dimensional random walk
- Equivalent to the Gambler's Ruin problem



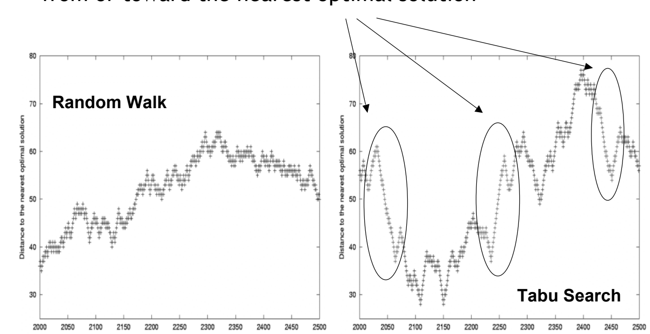
A Markov Model of Iterated Local Search

- A generalized one-dimensional random walk…
- … but with restricted transition probabilities
- Large-distance jumps are highly unlikely

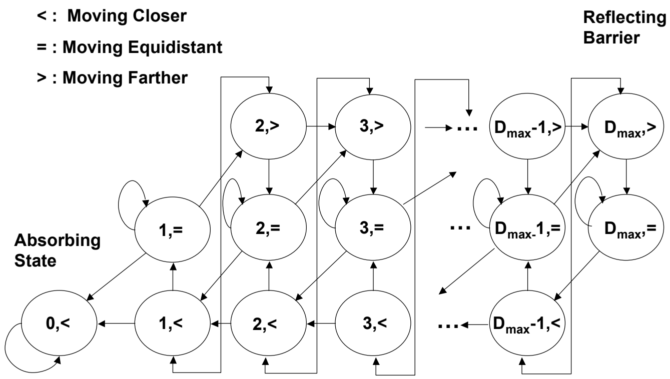


Short-Term Memory and the Dynamics of Tabu Search

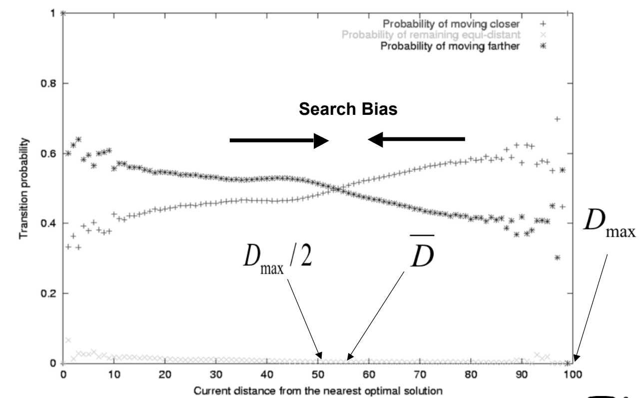
- Short-term memory consistently biases search either away from or toward the nearest optimal solution



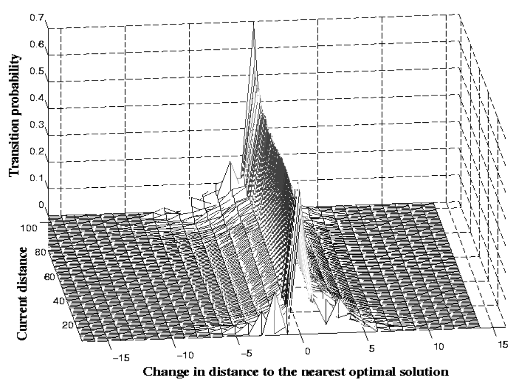
A Markov Model of Tabu Search



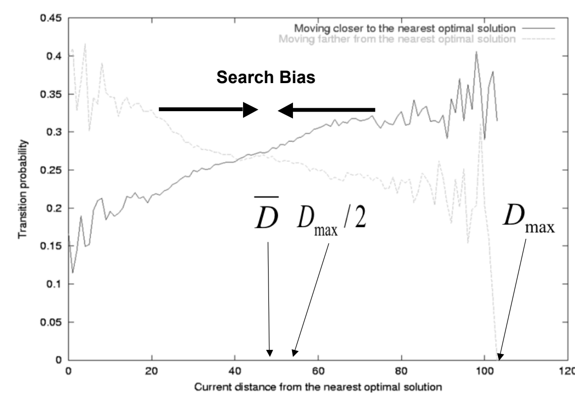
Transition Probabilities Under Metropolis Sampling



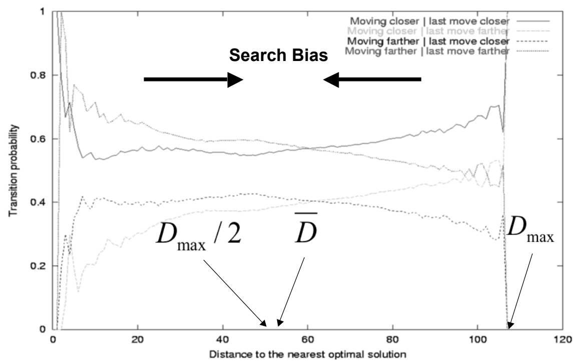
Transition Probabilities Under Iterated Local Search



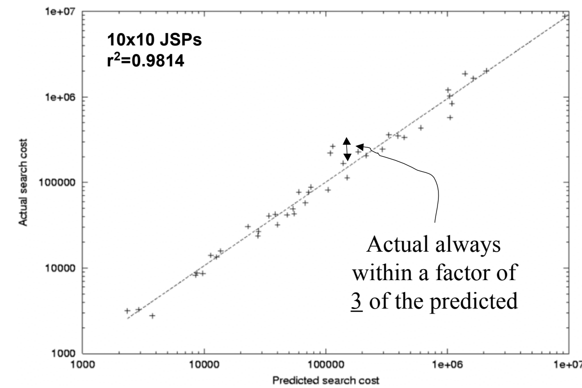
Transition Probabilities Under Iterated Local Search



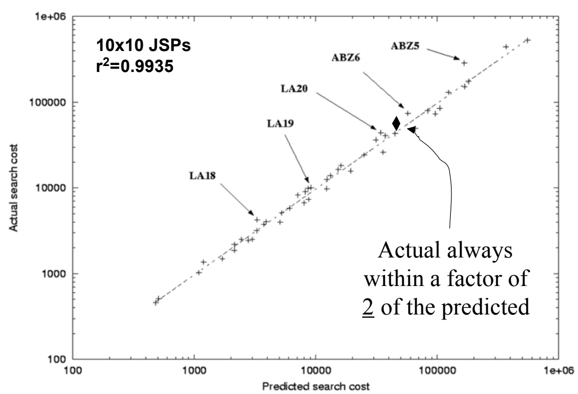
Transition Probabilities Under Tabu Search



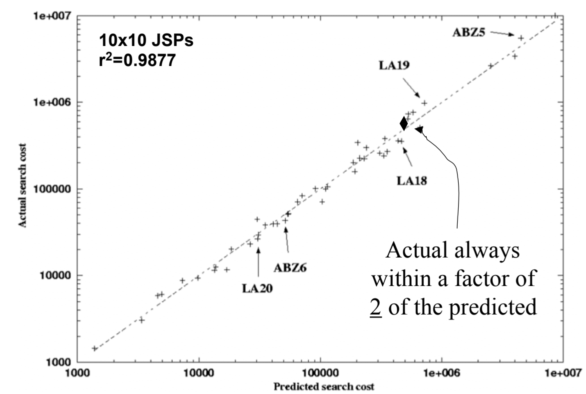
Dynamic Cost Model Accuracy: Metropolis Sampling



Dynamic Cost Model Accuracy: Iterated Local Search



Dynamic Cost Model Accuracy: Tabu Search



The Relationship Between the Cost Models

Search is biased toward solutions that are distance $\overline{D}$ from the nearest globally optimal solution

Search is biased toward solutions that are *approximately* distance $D_{\max}/2$ from the nearest globally optimal solution

=> $D_{\max} \approx 2\overline{D}$!

- $\overline{D}$ *estimates* a key parameter of the dynamic model
- The static and quasi-dynamic models provide increasingly accurate estimates of $\overline{D}$
- Implication: Landscape structure and run-time dynamics are tightly linked

Future Research Opportunities

- Generalization to other algorithms?
- Generalization to other problems?
- How does problem structure impact cost models?
- Applications
  - Can we estimate bias strength and D_{max}?
  - Can we *predict* search cost?
  - With what level of accuracy?
- Algorithm design
  - How can we minimize the impact of search space bias?
  - Do different representations induce different biases?
- *The analysis of fitness landscape structure and problem difficulty is effectively an open area*

Closing Thoughts

- Fitness landscape structure is a key determinant in problem difficulty for a wide range of algorithmic search paradigms
  - *Ignoring structure in algorithm design leads to "iterated hacking"*
- Many landscape features thought to be highly correlated with problem difficulty aren't
  - *Always test your hypotheses*
- There can be very clear relationships between fitness landscape structure and algorithm run-time behavior
  - But these can only be identified via careful experimentation and analysis
- This research area is largely open
  - A lot of papers sound conclusive, but if you look more closely…