AN INTEGRATED WHITE+BLACK BOX APPROACH FOR DESIGNING AND TUNING STOCHASTIC LOCAL SEARCH
In this thesis, we are dealing with COPs
- Combinatorial Optimization Problem
- Issue: NP-hardness of those COPs!

Types of algorithms to solve COPs:

<table>
<thead>
<tr>
<th>Optimality</th>
<th>Exact Solver</th>
<th>Non-Exact / Incomplete Solver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimality Guaranteed</td>
<td>Not Guaranteed</td>
<td></td>
</tr>
<tr>
<td>Run Time</td>
<td>Intractable in practice</td>
<td>Usually fast, user sets the bounds,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>‘anytime’ behavior</td>
</tr>
<tr>
<td>Example</td>
<td>B&amp;B, CP, IP, LP, etc</td>
<td>Approximation, Heuristic,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Metaheuristic/SLS, etc</td>
</tr>
</tbody>
</table>
Focus: Non-Exact SLS Algorithms

- Our focus: The non-exact Stochastic Local Search (SLS) (Metaheuristic) algorithms, examples:
  - Tabu Search (TS)
  - Iterated Local Search (ILS)
  - Simulated Annealing (SA)
  - Evolutionary Algorithm (EA)
  - Ants Colony Optimization (ACO), etc

- SLS “works well” in practice especially when:
  - Run time is ‘limited’
  - Instance size is ‘large’
  - ‘Slightly’ non-optimal results are acceptable
Outline

2. Some Quick Background

3. SLS Design and Tuning Problem
   • What is the problem?
   • What have been done?

4. White-Box FLST Visualization

5. SLS Visualization Tool: Viz

6. Integrated White+Black Box Approach

7. Case Studies

The problem addressed in this thesis, published in MIC05 & its Post-Conf book chapter
Motivation: Engineering SLS

- Typical situation:

  1. Pick SLS Algorithm
  2. Select SLS Configurations
  3. Implement SLS code
  4. Run SLS on COP Training Instances
  5. Analyze Performance on COP Test Instances

0. Have an (NP-)hard COP

NOT Straightforward!

We can keep going on forever if we keep aiming for faster and better SLS algorithm!
Anatomy of an SLS (1)

- The SLS algorithm template is typically simple, e.g.

- **TABU-SEARCH**
  - CurrentSolution = OverallBest = InitialSolution
  - while (terminating-condition-not-satisfied)
    - BestMove = Best([Neighborhood],[TabuMechanism], [AspirationCriteria],CurrentSolution)
    - CurrentSolution = BestMove(CurrentSolution)
    - [TabuMechanism].SetTabu(CurrentSolution,BestMove,TabuTenure)
    - if (Better(CurrentSolution,OverallBest))
      - OverallBest = CurrentSolution
    - if (Something_Happens())
      - Do_A_Strategy()
  - return OverallBest

Q: How to **tune** these parameter values?

Q: How to **design** these components and strategies?
Anatomy of an SLS (2)

There are many configurable parts in Tabu Search:

- **Type-1: Setting Tabu Tenure Length:**
  - Guessing/Trial and Error?
  - Using past experience as a guide?

- **Type-2: Selecting Local Neighborhood:**
  - 2/3/k-opt?
  - Very Large Scale Neighborhood (VLSN)?

- **Type-2: Selecting Tabu Mechanism and Aspiration Criteria:**
  - Tabu moves/attributes/solutions? Tabu list/table?
  - Use Aspiration Criteria? What are the criteria?

- **Type-3: Adding Search Strategies:**
  - Intensification versus Diversification?
  - Robust TS, Reactive TS?
  - Hybridization?
  - When & How to apply these strategies?

---

Different configuration $\approx$ Different SLS performance!

The Challenge: SLS DTP

- SLS **Design and Tuning Problem (DTP)** is a problem of:
  - Finding an SLS & its configuration (holistic) that has good performance on training and test instances under *limited budget*:
    - Limited Development Time (e.g. 1 week)
    - Limited Running Time (e.g. 10 minutes/SLS run, or 2s as in MiniProj1)

- Some Quotes:
  - Barr *et al.*, 1995
    - “The selection of parameter values that drive heuristics is itself a scientific endeavor, and deserves more attention than it has received in the Operations Research literature.”
  - Birattari, 2004
    - “For obtaining a fully functioning algorithm, a metaheuristic needs to be configured: typically some modules need to be instantiated and some parameters need to be tuned.”
  - Adenso Diaz & Laguna, 2006
    - “There is anecdotal evidence that about 10% of the total time dedicated to designing and testing of a new heuristic or metaheuristic is spent on development, and the remaining 90% is consumed (by) fine-tuning (its) parameters.”

- DTP is the **bottleneck** of SLS engineering!
  - How to address it?
Addressing DTP: **Black-Box**

(Halim & Lau, 2007) – Metaheuristics - Progress in Complex Systems Optimization

- **Black-Box Approach**
  - Human starts the SLS engineering process:
    - Design the SLS
    - Provide the initial configuration set
  - Machine usually the one doing the dirty job:
    - Explore the given configuration set
    - Do it systematically and as efficient as possible
    - Return the best found configuration
  - **Examples:**
    - ParamILS (Hutter *et al.*, 2007), [http://www.cs.ubc.ca/labs/beta/Projects/ParamILS/](http://www.cs.ubc.ca/labs/beta/Projects/ParamILS/)
  - For Mini Project, you can build a simple one yourself, i.e. design a (small) set of parameter values to be tried and systematically try them one by one and use the best one…
White-Box Approach

- Exploit human intelligence:
  - Analyze the SLS behavior
  - (Re-)design or tweak the SLS
- Machine helps the analysis part:
  - Help human in analyzing the SLS runs
  - Either via statistical measures or visualization
- Examples:
  - Statistical Analysis, e.g. FDC, RTD (Jones, 1995; Merz, 2000; Hoos & Stuetzle, 2005)
  - Human Guided Tabu Search (Klau et al., 2002)
  - Visualization of Search Process (Syrjakow & Szczerbicka, 1999; Kadluczka et al., 2004)
  - V-MDF and Viz (Lau et al, 2005; Halim et al., 2006a/b; Halim & Lau, 2007; Halim & Yap 2007)

Note: our first five publications deal with mainly white-box visualization approach only
No Clear Winning Solution Yet

Only deal with parameter tuning
Deal with choosing components
Deal with designing search strategies

<table>
<thead>
<tr>
<th>Approach</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type of Method</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>B</td>
<td>W</td>
<td>W</td>
<td>W</td>
</tr>
<tr>
<td>Addressing type-1</td>
<td>H</td>
<td>M</td>
<td>M</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>Addressing type-2</td>
<td>H</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>E</td>
<td>E</td>
<td>H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>Addressing type-3</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>¬</td>
<td>H</td>
<td>M</td>
<td>E</td>
</tr>
<tr>
<td>Ease of Usage</td>
<td>H</td>
<td>M</td>
<td>H</td>
<td>E</td>
<td>M</td>
<td>M</td>
<td>H</td>
<td>M</td>
<td>M</td>
</tr>
</tbody>
</table>

Comparison of the Reviewed Approaches

Approaches:
A: Blind Tuning, B: Brute Force Tuning, C: Meta SLS,
D: CALIBRA, E: F-Race, F: ParamILS,
G: Statistical Analysis, H: Human-Guided Search,
I: Visualization of Search Behavior,

Legends:
B: Black-box, W: White-box, W+B: Integrated Approach,
E: Easy, M: Medium, H: Hard, ¬: Not applicable.
Outline: Our Contributions

2. Some Quick Background
3. SLS Design and Tuning Problem
4. White-Box FLST Visualization
   - White-Box Analysis with Visualizations
   - FLST Visualization Ideas
5. SLS Visualization Tool: Viz
6. Integrated White+Black Box Approach
7. Case Studies

Our proposed approach (in all our papers), especially SLS07
Our visualization tool, published in ECAI06 & UIST06
Our integrated approach, published in CP07
Our results (in all our papers)
Analysis of SLS Algorithm

- If SLS algorithm does not have ‘desirable performance’ yet, the algorithm designer needs to analyze it!
  - We need **insights** on how to further improve the algorithm!
  - There are many SLS behavior/trajectory questions to be answered!
- Two common white-box approaches:
  - Statistical Methods
    - Solution Quality Distribution (Robustness)
    - Speed/Runtime/Iteration or Run Time Distribution (RTD)
    - Fitness Distance Correlation (FDC) Analysis
  - Information Visualization (**we will use this**)  
    - Solution Quality over Time
    - Visualization of Fitness Landscape, Search Trajectories, etc
Exploiting Human Strengths
(We can easily answer some questions that computer cannot do well at present)

- Try to read the words in these figures:
- Find a word that relates to all the images:

Information Visualization
(Maximizing the power of Human Brain to discern large data)

The cause of Global Warming?

Data taken from http://www.inference.phy.cam.ac.uk/mackay/presentations/SEWTHA6s/index.html
Existing SLS Visualizations

- Objective Value over Time (Solution Quality Distribution)
- Fitness Distance Correlation (Jones, 1995; Watson et al., 2005)
- Problem Specific, e.g. TSP (Klau et al., 2002)
- 2-D Animation of 2-D Problem (Syrjakow & Szczerbicka, 1999)
- N-to-2-Space Mapping (Kadluczka, 2004)
- Each of these visualization cannot show search trajectory well, but we can do better…
An SLS Visualization Technique

**FLST Visualization – The Ideas**

Objectives of FLST visualization:

- Must be a generic visualization!
  - How to achieve it?
- Must be able to assist user to gain insights!
  - Insights to match SLS Trajectory with COP Fitness Landscape!
  - Issues:
    - Same COP, different instance, may have different FL
    - Same SLS, different FL, may have different behavior
    - Many details (e.g. SLS configuration) determines the actual ST
    - SLS is stochastic, every run is different
    - How to visualize exponentially large FL?
    - How to visualize ST on FL?
Fitness Landscape Search Trajectory (FLST) (1)
Generic Visualization

- **Basic Concepts:**
  - **Fitness Landscape** ~> abbreviated as FL
  - **Search Trajectory** ~> abbreviated as ST
  - **Anchor Points** ~> abbreviated as AP
  - FLST Visualization
Fitness Landscape Search Trajectory (FLST) (2)

Generic Visualization

- **Basic Concepts:**
  - **Fitness Landscape**
  - **Search Trajectory**
  - **Anchor Points**
  - FLST Visualization

\[
FL(\pi) = \langle S(\pi), d(s_1, s_2), g(\pi) \rangle
\]

- \( S(\pi) \): set of solutions of the COP instance
- \( g(\pi) : S \rightarrow \mathbb{R} \): objective function
- \( d(s_1, s_2) : S \times S \rightarrow \mathbb{R} \): distance function
Fitness Landscape Search Trajectory (FLST) (3)

Generic Visualization

Basic Concepts:

- **Fitness Landscape**
- **Search Trajectory**
- **Anchor Points**
- **FLST Visualization**

\[ \text{ST} = \{s_0, s_1, \ldots, s_k\} \in S(\pi) \]

\[ \forall i \in \{1, 2, \ldots, k\}, (s_{i-1}, s_i) \text{ is either a local move according to } N(\pi) \]

or a stronger diversification move.
Fitness Landscape Search Trajectory (FLST) (4)
Generic Visualization

Basic Concepts:
- Fitness Landscape
- Search Trajectory
- Anchor Points
- FLST Visualization

Note: we may still miss some good points
Fitness Landscape Search Trajectory (FLST) (5)

Generic Visualization

- Basic Concepts:
  - Fitness Landscape
  - Search Trajectory
  - Anchor Points
  - FLST Visualization
    - FLO Mode

Abstract 2-D space

Use appropriate distance metric
Fitness Landscape Search Trajectory (FLST) (6)

Generic Visualization

- **Basic Concepts:**
  - Fitness Landscape
  - Search Trajectory
  - Anchor Points
  - FLST Visualization
    - **FLO Mode**
      - Distance Metric

\[ d(s_1, s_2) : S \times S \to R \]

- **Hamming Distance** → QAP
  \[ d(s_1, s_2) : \text{mismatches between } s_1 \text{ & } s_2 \]
  \[ d(\{1,2,3,4,5\}, \{1,2,3,4,5\}) = 0 \]
  \[ d(\{1,2,3,4,5\}, \{4,1,2,3,5\}) = 4 \]

- **Hamming Distance** → LABS
  \[ d(s_1, s_2) : \text{mismatches between } s_1 \text{ & } s_2 \]
  \[ d(\{1,1,-1,-1,1\}, \{1,1,-1,-1,1\}) = 0 \]
  \[ d(\{1,1,-1,-1,1\}, \{1,1,1,-1,-1\}) = 2 \]

- **Bond Distance** → TSP
  \[ d(s_1, s_2) : \text{different edges between } s_1 \text{ & } s_2 \]
  \[ d(\{1-2-3-4-5\}, \{1-2-3-4-5\}) = 0 \]
  \[ d(\{1-2-3-4-5\}, \{4-1-2-3-5\}) = 2 \]
Fitness Landscape Search Trajectory (FLST) (7)

Generic Visualization

- **Basic Concepts:**
  - **Fitness Landscape**
  - **Search Trajectory**
  - **Anchor Points**
  - **FLST Visualization**
    - **FLO Mode**

- **Color and shape indicates solution quality:**
  - **Blue Circle:** Good
  - **Green Triangle:** Medium
  - **Yellow Rectangle:** Bad
  - **Black Dot:** Very Bad

- **Abstract 2-D space**
  - Use appropriate distance metric
  - Generic!
Fitness Landscape Search Trajectory (FLST) (8)
Generic Visualization

- **Basic Concepts:**
  - **Fitness Landscape**
  - **Search Trajectory**
  - **Anchor Points**
  - **FLST Visualization**
    - FLO Mode
    - ST example 1

Insight about Pink Trajectory:
Solution Cycling in bad black/yellow APs. Fail to reach better green/blue APs.
Basic Concepts:

- **Fitness Landscape**
- **Search Trajectory**
- **Anchor Points**
- **FLST Visualization**
  - FLO Mode
  - ST example 1
  - ST example 2

Insight about Pale Blue Trajectory:
Diversify after hitting an AP (local optima).
Manage to reach better green/blue APs
FLST can roughly explain:

Fitness Landscape Structures

1. Distribution of Local Optima
   1. Clustered
   2. Spread

2. Variance of Local Optima
   1. Rugged
   2. Smooth
   3. Isolated

Search Trajectory Behaviors

1. Behave as intended?
2. Intensification ok?
3. Diversification ok?
4. Any cycling behavior?
5. Where in FL does the SLS search?
6. How does the SLS make progress?
7. Initial solution versus BK solution?
8. How wide is the SLS coverage?
9. Effect of modifying configuration?
10. SLS comparison?

Behavior of a heuristic and stochastic SLS is "complex"

But with this novel FLST visualization, one can approximately check whether his SLS performs well on the corresponding FL…
Outline

2. Some Quick Background
3. SLS Design and Tuning Problem
4. White-Box FLST Visualization
5. **SLS Visualization Tool: Viz**
   - Screen Shots
   - User Interface Features of Viz
6. Integrated White+Black Box Approach
7. Case Studies

**Our visualization tool, published in ECAI06 & UIST06**

**Our integrated approach, published in CP07**

Our results (in all our papers)
Evolution of Viz (2006-2008)
Two Programs in Viz: EW & SIMRA

Experiment Wizard & Single Instance Multiple Runs Analyzer

Please visit: www.comp.nus.edu.sg/~stevenha/viz or sls.visualization.googlepages.com

Will be explained later
Viz’s Technical Notes

- Viz is created using:
  - Microsoft Visual C# .NET 2005
  - Microsoft .NET Framework 2.0
  - CsGL (C# wrapper for OpenGL)
  - OpenGL 1.2 (Graphics Library)

- SLS algorithms are implemented using:
  - Microsoft Visual C++ .NET 2005

- Viz version 3.2008.11.13 executable that I packaged back in 2009/2010 can NOT run on Windows 8 or 10 😞....
Viz UI Features (1)

- **Multi-Source Visualizations**
- Visual Comparison
- Animated Search Playback
- Color and Highlighting
- Multiple Level of Details
- Textual Data

**Playback SLS runs from several angles:**

- **Generic Visualizations**
  - Fitness Landscape Search Trajectory
  - Objective Value
  - Fitness Distance Correlation
  - Event Bar

- Algorithm Specific Visualization
- Problem Specific Visualization

- Coordinated Visualizations
Viz UI Features (2)

- Multi-Source Visualizations
- Visual Comparison
- Animated Search Playback
- Color and Highlighting
- Multiple Level of Details
- Textual Data

Exploit human comparison abilities
- Not a perfect judge
- But, good at relative comparison
  - w.r.t a given baseline

Useful for understanding:
- **Behavior:**
  - Same SLS & configuration, different iterations
- **Robustness:**
  - Same SLS & configuration, two runs
- **Effect of Modifications:**
  - Same SLS, different configurations
- **Overall behavior:**
  - Two totally different SLS and configurations
Viz UI Features (3)

- Multi-Source Visualizations
- Visual Comparison
- **Animated Search Playback**
- Color and Highlighting
- Multiple Level of Details
- Textual Data

SLS trajectory information is too large:
- Information overload if presented all once
- Animation over time show more details
Viz UI Features (4)

- Multi-Source Visualizations
- Visual Comparison
- Animated Search Playback
- **Color and Highlighting**
- Multiple Level of Details
- Textual Data

- **Color is our main highlighting tool**
  - Utilize human’s pre-attentive visual feature
    - For Labeling / Categorizing:
      - Run 1: **Red**, Run 2: **Blue**
      - AP Quality: **Good/Medium/Bad/VeryBad**

- SLS typically runs in thousand iterations…
  - Analyzing from start to end is overwhelming

- Event Bar highlights interesting regions
  - Viz computes highlights/index points for generic events (e.g. new best found solution)

New best found solutions...  Current solution + some information
Viz UI Features (5)

- Multi-Source Visualizations
- Visual Comparison
- Animated Search Playback
- Color and Highlighting
- Multiple Level of Details
- Textual Data

Different level of detail is required for different purpose

Zoom in

Zoom out
Viz UI Features (6)

- Multi-Source Visualizations
- Visual Comparison
- Animated Search Playback
- Color and Highlighting
- Multiple Level of Details
- **Textual Data**

We add textual and statistical data to further elaborate the visualization!
Outline

2. Some Quick Background ✓
3. SLS Design and Tuning Problem ✓ ✓
4. White-Box FLST Visualization ✓ ✓ ✓
5. SLS Visualization Tool: Viz ✓ ✓ ✓
6. Integrated White+Black Box Approach
   • White-Box Alone Is Not Enough
   • Our Integrated Approach
7. Case Studies

Our integrated approach, published in CP07

Our results (in all our papers)
Integrated Approach
(Illustration on Shortest Path Problem)

- **Black-box Approach**
  - **Pro:**
    - Simple to apply
  - **Cons:**
    - If solution is out of the box?
    - If configuration set is too big?

- **White-box Approach**
  - **Pro:**
    - Obtain Insights
  - **Cons:**
    - Fine-tuning is tedious

**Solution: Integrate them!**
- Combine their strengths
Man-Machine Interaction

Human
- Intelligence
- Visual Capability
- Innovative
- Common Sense
- etc

Computer
- Speed
- Reliable
- Endurance
- Unbiased
- etc

Bridging Interface: Information Visualization

Tasks:
Design the SLS, analyze and understand the SLS, re-design the SLS if necessary.

Tasks:
Run SLS, visualize search, compute statistics, do black-box fine tuning.
Basic Ideas

- **White-box: FLST Visualization**
  - Opens up the ‘box’
  - Analyze Fitness Landscape (FL)
  - Analyze Search Trajectory (ST) on FL
  - Insights to (re)design the SLS
  - Narrow down the configuration set

- **Black-box: Fine Tuning**
  - Get the ‘smaller’ configuration set
  - Fine-tune the focused configuration set

**Example:**
- (Rugged)
- (Stuck in LO)
- (More Diversification)
- (Less Intensification)
- (Focus on Diversification)
- (How much Diversification?)
Integrated Approach

(A flow chart that summarizes the entire SLS engineering process)

1. Instances separation (training/test set)
2. Create initial (pilot) SLS for the COP
3. Fitness Landscape Overview Analysis
4. Split the instances into several classes (if necessary)
5. Formulate hypotheses of effective walks for each class
6. Understand and Tune the SLS to behave as intended
7. Use ‘Black-Box’ Tuning Algorithm (if needed)
8. How is the SLS behavior on training instances?
   - Mismatch Hypothesis Poor Performance
   - Match Hypothesis Poor Performance
   - Mismatch Hypothesis Good Performance
   - Match Hypothesis Good Performance
9. Are the results on test instances good?
   - Yes
   - No

Done
Viz EW Features for IWBBBA

- Label A, B, C: Problem, Algorithm, & Experiment Design
- Label D: Presentation of experiment results
- Built-in support for computing descriptive and inferential statistics
- Built-in support for doing simple black-box tuning algorithm
IWBBBA Using Viz
Outline

2. Some Quick Background ✓
3. SLS Design and Tuning Problem ✓
4. White-Box FLST Visualization ✓
5. SLS Visualization Tool: Viz ✓
6. Integrated White+Black Box Approach ✓
7. Case Studies
   - ILS for TSP (until Week 13)
   - Ro-TS for QAP (2 fitness landscapes → 2 SLS)
   - TS for LABS (until Week 13, state-of-the-art results!!)

Our experimental results support our proposed approach
Scenario 2:
Objectives: Show that COP can have >1 FL types and algorithm specialization is needed in ‘short run’ setting

Robust Tabu Search for Quadratic Assignment Problem

For details, see:
Problem Description

- **QAP**: Quadratic Assignment Problem

- **Input**:  
  - $n$ – the size of matrix $A$ and $B$.  
  - matrix $A$ of size $n \times n$.  
  - matrix $B$ of size $n \times n$.

- **Problem**:  
  - Find an assignment (permutation) $\pi$ that minimizes the objective function:

$$Z(A,B,\pi) = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{i,j} \pi_{i} \pi_{j} b_{i,j}$$

- **Output**:  
  - The best assignment.
QAP is NP-hard

- QAP is NP-hard
- Let A be all zeroes but cells that connect two indices i and j that differ by 1 (modulo n) are given values 1
- Let B be the original TSP distance matrix

<table>
<thead>
<tr>
<th>A</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>20</td>
<td>42</td>
<td>35</td>
</tr>
<tr>
<td>1</td>
<td>20</td>
<td>0</td>
<td>30</td>
<td>34</td>
</tr>
<tr>
<td>2</td>
<td>42</td>
<td>30</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>3</td>
<td>35</td>
<td>34</td>
<td>12</td>
<td>0</td>
</tr>
</tbody>
</table>

- The output of this QAP instance is the solution of the TSP instance so we have just reduced NP-hard TSP into QAP
- So QAP is also NP-hard
I built a mini tool for this

See QAP_FL_Demo.exe

Optimal TSP answer
Fitness Landscape Analysis

Fitness Landscape of tai30a, similar as lipa20a/tai35a/wil50, etc
Observation (A): spread, smooth

Fitness Landscape of tai30b, similar as kra30a/ste36a/tai35b, etc
Observation (B): spread, rugged

Reasonable hypothesis: QAP has at least 2 fitness landscape types!
Ro-TS-I/A for Type A Instances

Seems ok but can be improved.
Tabu tenure range [90-110]%*n.

Ro-TS-I

Ideal Walk

Ro-TS-A

Smaller tabu tenure range for more intensification [40-80]%*n.

<table>
<thead>
<tr>
<th>Training Instance</th>
<th>Best Known</th>
<th>Ro-TS-I (M,s) Off</th>
</tr>
</thead>
<tbody>
<tr>
<td>tai30a</td>
<td>1.818.146</td>
<td>0.92% (0.22)</td>
</tr>
<tr>
<td>tai35a</td>
<td>2.422.002</td>
<td>1.13% (0.39)</td>
</tr>
<tr>
<td>tai50a</td>
<td>4.938.796</td>
<td>1.72% (0.12)</td>
</tr>
</tbody>
</table>
Ro-TS-I/B for Type B Instances

Stuck in a deep local optima

More partial diversifications to explore different parts of FL!

<table>
<thead>
<tr>
<th>Training Instance</th>
<th>Best Known</th>
<th>Ro-TS-I (M,s) Off</th>
</tr>
</thead>
<tbody>
<tr>
<td>tai30b</td>
<td>637.117.113</td>
<td>12.12 % (6.65)</td>
</tr>
<tr>
<td>tai35b</td>
<td>283.315.445</td>
<td>8.10 % (3.89)</td>
</tr>
<tr>
<td>tai50b</td>
<td>458.821.517</td>
<td>6.02 % (3.33)</td>
</tr>
</tbody>
</table>
### Results on Test Instances

<table>
<thead>
<tr>
<th>Instance</th>
<th>Iters</th>
<th>BK</th>
<th>OV</th>
<th>$\bar{x}$</th>
<th>$\sigma$</th>
<th>$\bar{x}$</th>
<th>$\sigma$</th>
<th>$\bar{x}$</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>lipa50a</td>
<td>12500</td>
<td>62093</td>
<td></td>
<td>0.17</td>
<td>0.35</td>
<td>0.04</td>
<td>0.18</td>
<td>0.81</td>
<td>0.28</td>
</tr>
<tr>
<td>lipa90a</td>
<td>40500</td>
<td>360630</td>
<td></td>
<td>0.46</td>
<td>0.16</td>
<td>0.44</td>
<td>0.15</td>
<td>0.58</td>
<td>0.02</td>
</tr>
<tr>
<td>rou20</td>
<td>2000</td>
<td>725522</td>
<td></td>
<td>0.24</td>
<td>0.16</td>
<td>0.14</td>
<td>0.14</td>
<td>0.26</td>
<td>0.16</td>
</tr>
<tr>
<td>tai25a</td>
<td>3125</td>
<td>1167256</td>
<td></td>
<td>1.05</td>
<td>0.40</td>
<td>1.04</td>
<td>0.40</td>
<td>1.28</td>
<td>0.29</td>
</tr>
<tr>
<td>tai40a</td>
<td>8000</td>
<td>3139370</td>
<td></td>
<td>1.23</td>
<td>0.29</td>
<td>1.23</td>
<td>0.22</td>
<td>1.74</td>
<td>0.29</td>
</tr>
<tr>
<td>tai60a</td>
<td>18000</td>
<td>7205962</td>
<td></td>
<td>1.62</td>
<td>0.17</td>
<td>1.51</td>
<td>0.19</td>
<td>2.13</td>
<td>0.24</td>
</tr>
<tr>
<td>wil100</td>
<td>50000</td>
<td>273038</td>
<td></td>
<td>0.20</td>
<td>0.08</td>
<td>0.18</td>
<td>0.09</td>
<td>0.16</td>
<td>0.04</td>
</tr>
</tbody>
</table>

### Type B Test Instances

<table>
<thead>
<tr>
<th>Instance</th>
<th>Iters</th>
<th>BK</th>
<th>OV</th>
<th>$\bar{x}$</th>
<th>$\sigma$</th>
<th>$\bar{x}$</th>
<th>$\sigma$</th>
<th>$\bar{x}$</th>
<th>$\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td>bur26b</td>
<td>3380</td>
<td>3817852</td>
<td></td>
<td>0.39</td>
<td>0.24</td>
<td>0.42</td>
<td>0.25</td>
<td>0.04</td>
<td>0.06</td>
</tr>
<tr>
<td>kra30b</td>
<td>4500</td>
<td>91420</td>
<td></td>
<td>0.30</td>
<td>0.39</td>
<td>0.55</td>
<td>0.69</td>
<td>0.24</td>
<td>0.22</td>
</tr>
<tr>
<td>scr20</td>
<td>2000</td>
<td>110030</td>
<td></td>
<td>0.35</td>
<td>0.64</td>
<td>0.81</td>
<td>1.14</td>
<td>0.10</td>
<td>0.22</td>
</tr>
<tr>
<td>ste36b</td>
<td>6480</td>
<td>15852</td>
<td></td>
<td>2.87</td>
<td>3.30</td>
<td>3.74</td>
<td>2.77</td>
<td>0.53</td>
<td>0.74</td>
</tr>
<tr>
<td>tai25b</td>
<td>3125</td>
<td>344355646</td>
<td></td>
<td>17.03</td>
<td>10.19</td>
<td>17.97</td>
<td>9.43</td>
<td>0.13</td>
<td>0.16</td>
</tr>
<tr>
<td>tai40b</td>
<td>8000</td>
<td>637250948</td>
<td></td>
<td>10.00</td>
<td>4.43</td>
<td>10.35</td>
<td>4.25</td>
<td>0.10</td>
<td>0.24</td>
</tr>
<tr>
<td>tai60b</td>
<td>18000</td>
<td>608215054</td>
<td></td>
<td>7.34</td>
<td>3.49</td>
<td>8.03</td>
<td>3.25</td>
<td>0.15</td>
<td>0.10</td>
</tr>
</tbody>
</table>

Table 7.8: Ro-TS-I/A/B Results on Test Instances (20 replications per run)
Conclusions, we have:

3. Identified SLS Design and Tuning Problem (DTP) as the issue to be addressed in this thesis
4. Created a generic Fitness Landscape Search Trajectory (FLST) visualization to analyze SLS behavior on COP
5. Build an SLS visualization tool: Viz with its UI features
6. Proposed the Integrated White+Black Box Approach (IWBBA) to address SLS DTP
7. Applied IWBBA using Viz on three SLS DTP scenarios successfully: ILS $\rightarrow$ TSP, Ro-TS $\rightarrow$ QAP, TS $\rightarrow$ LABS

Online version of this ppt: [http://www.comp.nus.edu.sg/~stevenha/viz](http://www.comp.nus.edu.sg/~stevenha/viz)

Thank you for your attention 😊
Related Publications
(In Reverse Chronological Order)


