UNIVERSITAT POMPEU FABRA

From ILS to Hybrid ILS …

and other extensions

Helena Ramalhinho Lourenço
lena.upf.edu
Universitat Pompeu Fabra Barcelona

ILS 1

Outline

- Introduction to ILS
- Applications of ILS
- Hybrid ILS and other Extensions
  - Hydrid with other metaheuristics
  - SimILS
  - Two-stage Optimization using ILS
  - MathILS

ILS 2

Outline

- Applications:
  - Market Basket Analyisis
  - BonArea
  - Supply Chain Design for ecommerce
  - SEAT/Volkwagen
  - Zara (INDITEX & OESIA)

ILS 3

Local optimization algorithms

- Local search
  - 1. Get a initial solution x (current solution). Use a constructive heuristic.
  - 2. Search the neighborhood. While there is an untested neighbor of x:
    - * 2.1 Let x' be an untested neighbor of x;
    - * 2.2 If c(x')<c (x) set x=x'; (x' is the new current solution)
  - 3. Return x (local optimal solution).

ILS 4

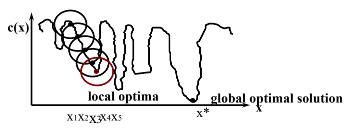
Local optimization algorithms

- Design of a local optimization algorithm:
  - Obtain an **initial solution**
    - * Heuristic
    - * Random solution
  - Define the **neighborhood**
    - * Specific for each problem
  - How to **search the neighborhood**
    - * Complete search
    - * First improvement

ILS 5

Local optimization algorithms

- Comments
  - The search stops at the first local optimum solution with respect to the neighborhood N.
  - The final solution highly depends on the initial solution and on the neighborhood.
  - No way back out of unattractive local optima...



ILS 6

UNIVERSITAT POMPEU FABRA

Multi-Start

- Iterative improvement or hill-descending
  - 1. Get a initial random solution x.
  - 2. Run an **local optimization** (output x)
  - 3. If cost(x)<cost(x_{best}) set x_{best}=x;
  - 4. If the stop criteria is not verified, go back to step 1.
  - 5. Output the best solution found.
    - Comments
      - Successive repetition of local improvement.
      - Easy to implement.
      - Random solutions may be very bad.

ILS 7

Iterated Local Search

- A Local Search Method…
  - Single chain on search
- Search on the space of local optimal solutions
- Combines local optimization with a big transition/large step/perturbation.
  - Perturbation should not be easily undone by the local search
  - Most important aspect of the ILS
- Able to make large changes at any stage of the algorithm.

ILS 8

Iterated Local Search

- Get an **initial solution** x;
  - Heuristic method or a random solution.
  - **local optimization** method
- For a certain number of iterations:
  - **Perturbation Step**
    - method that makes a large modification based in optimization and on the structure of the solution x, resulting in x'.
  - Small-steps
    - **local optimization** method, initial solution x'; final solution x''.
  - **Perform an accept/reject test**
    - accept all solutions, accept with a certain probability or accept only if it is a better solution.
    - If x'' is accepted, then x = x''.
- Return the best solution found.

ILS 9

Iterated Local Search

- A simple implementation…
  - **Generate Initial Solution**
    - Greedy Heuristic
  - **Local Search Method**
    - First improvement local search
    - Definition of neighborhood
  - **Perturbation Method**
    - One move of a high level neighborhood
  - **Acceptance Criteria**
    - Accept if a better solution is found

Often leads to very good performance

Only requires few lines of additional code

ILS 10

Iterated Local Search

- **Generate Initial Solution**
  - Randomized Greedy Heuristic
  - Random solution
- **Acceptance Criteria**
  - Better
  - Random Walk
  - Simulated Annealing type
  - Restart
- **Local Search Method**
  - Local search
  - Tabu search
- **Perturbation**
  - Higher level of neighborhood
  - Strength of the perturbation
    - Big/small
  - Adaptive memory
  - Modify input data
  - Optimized perturbation

ILS 11

Iterated Local Search

- Improving ILS
  - **Relationship between local search and perturbation**.
    - Perturbation must lead to a new region of the solution space that cannot be reached by a local search method.
    - Perturbation should not be easily undo by the local search.
  - Perturbation can incorporate problem-specific information.
    - As for example optimization methods
    - Destruction and construction approach
  - A good perturbation transforms one excellent solution into a excellent starting point to a local search.
  - Local search method must be fast.
  - Complexity must be added progressively and in a modular way.

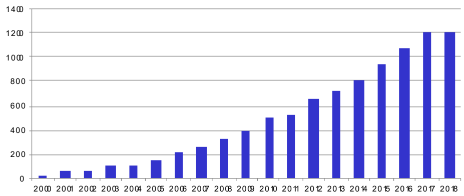
ILS 12

UNIVERSITAT POMPEU FABRA

Universitat Pompeu Fabra Barcelona

Iterated Local Search

- Google Scholar's number of publications
- **"Iterated Local Search"**
  - About 10,000 publications



ILS 13

Universitat Pompeu Fabra Barcelona

Iterated Local Search

- ILS applied to Complex and large-scale real problems

Accuracy	• Complex problems. • Large scale problems.
Speed	• Fast answer. • Analysis of several scenarios.
Flexibility	• Fast changes. • Different constraints in different areas.
Simplicity	• Need of fast implementation

ILS 14

Universitat Pompeu Fabra Barcelona

Traveling Salesman Problem

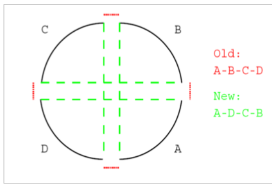
- Traveling Salesman Problem
  - Given a number of cities and the costs (distances) of traveling from any city to any other city…
  - What is the least-cost round-trip route that visits each city exactly once and then returns to the starting city?
  - http://www.math.uwaterloo.ca/tsp/

ILS 15

Universitat Pompeu Fabra Barcelona

Traveling Salesman Problem

- Generate Initial Solution
  - Constructive Heuristic: nearest neighbor, insertion heuristic
- Local Search Method
  - 2-opt/3-opt Neighborhood
- Perturbation Method
  - One 4-opt move (double-bridge)
- Acceptance Criteria
  - Accept only if the best solution improved

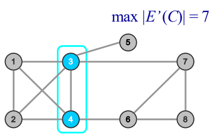


ILS 16

Universitat Pompeu Fabra Barcelona

Maximum Cut-Clique Problem

- Given a clique C, its edge neighborhood (cut-clique) is defined by the set of edges $E'(C)=\{(i,j) \in E: i \in C \text{ and } j \in V \setminus C\}$, and $|E'(C)|$ is its size. Denote $N(i)=\{j \in V:(i,j) \in E\}$.
- Maximum Cut-Clique
  - Maximum edge neighborhood clique



ILS 17

Universitat Pompeu Fabra Barcelona

ILS for Cut-Clique Problem

- Perturbation
  - Select randomly one node
  - Build the clique with all nodes in the previous clique and fully connected to this node
    Set $C \leftarrow [C \cap N(i)] \cup \{i\}$;
    Set $U \leftarrow \emptyset$ and $C' \leftarrow C$;
- Local Optimization
  - Add, Swap and Aspiration moves
  - R-ILS (random version)
  - D-ILS (deterministic version)

ILS 18

UNIVERSITAT POMPEU FABRA

Universitat Pompeu Fabra Barcelona

Maximum Cut-Clique Problem

- Traditional approach
  - Solve with Integer Linear Programming Software
  - Branch-and-Bound / Branch-and-cut general exact algorithm
  - Obtain the Optimal Solution / Lower Bound
  - CPLEX Optimizer

Universitat Pompeu Fabra Barcelona

Computational results:

Intel Core i7-2600 with 3.40GHz and 8GB RAM; using CPLEX 11.2

The ---- symbol means that CPLEX was not able to read and preprocess the model in one hour

time in seconds

Instance	$\|V\|$	$\|E\|$	d	MC problem		MCC Problem		
				$\omega(G)$	$\|N(C)\|$	$\|C\|$	$\|N(C)\|$	time
d1-RTN	2418	9317	0.0032	10	195	8	1273	605.11
d3-RTN	4755	26943	0.0024	18	1097	----	----	----
d7-RTN	6511	44615	0.0021	18	1576	----	----	----
d15-RTN	7965	62136	0.0020	18	1979	----	----	----
d30-RTN	10101	91803	0.0018	21	13099	----	----	---
d66-RTN	13308	148035	0.0017	----	----	----	----	----
c-fat200-1	200	1534	0.077	12	72	9	81	0.05
c-fat200-2	200	3235	0.163	24	264	17	306	0.09
c-fat200-5	200	8473	0.426	58	1682	44	1892	0.05
c-fat500-1	500	4459	0.036	14	98	11	110	0.76
c-fat500-2	500	9139	0.073	26	338	19	380	0.80
c-fat500-5	500	23191	0.186	64	2048	48	2304	0.83
c-fat500-10	500	46627	0.374	126	7938	94	8930	0.58

Universitat Pompeu Fabra Barcelona

Computational results of the ILS:

Intel Core i7-2600 with 3.40GHz and 8GB RAM

100 runs

time in seconds

Instance	$\|V\|$	$\|E\|$	d	MCC Problem		
				$\|C\|$	$\|N(C)\|$	time
d1-RTN	2418	9317	0.0032	8	1273	0.1762
d3-RTN	4755	26943	0.0024	12	3526	0.4743
d7-RTN	6511	44615	0.0021	15	5656	0.6777
d15-RTN	7965	62136	0.0020	16	7772	0.8757
d30-RTN	10101	91803	0.0018	21	13099	1.1317
d66-RTN	13308	148035	0.0017	28	22379	1.4081
c-fat200-1	200	1534	0.077	9	81	0.1385
c-fat200-2	200	3235	0.163	17	306	0.0866
c-fat200-5	200	8473	0.426	44	1892	0.0664
c-fat500-1	500	4459	0.036	11	110	0.5451
c-fat500-2	500	9139	0.073	19	380	0.3595
c-fat500-5	500	23191	0.186	48	2304	0.2381
c-fat500-10	500	46627	0.374	94	8930	0.2111

Universitat Pompeu Fabra Barcelona

Market Basket Analysis

- The main objective is to analyze large dataset of store transactions
- Obtain relevant insights to do a better planning of the Marketing strategies and operations.
  - Product placement
  - Optimal product-line offering
  - Personalized marketing campaigns
  - Product promotions

Universitat Pompeu Fabra Barcelona

Market Basket Analysis

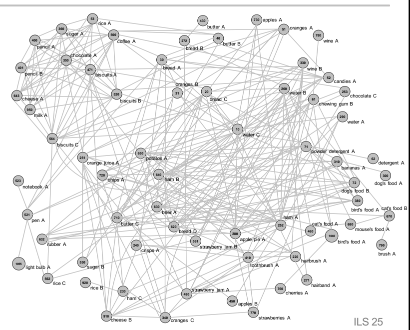
- Raeder and Chawla (2011) say "... no techniques currently available in the literature sufficiently addresses the problem of finding meaningful relationships in a large transaction databases."
- Dataset
  - a household panel database for the British ice cream market.
  - 691 different *varieties of products* available in the British market.

Universitat Pompeu Fabra Barcelona

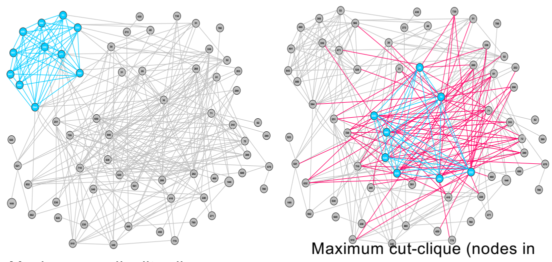
Market basket network

nodes: products in a store

edges: represent pairs of products (*i*,*j*) bought together by a customer on a given purchase visit to the store



Market basket network



Maximum cardinality clique (nodes in blue)

Maximum cut-clique (nodes in blue in red represent threpresent the clique, edges e cut-clique)

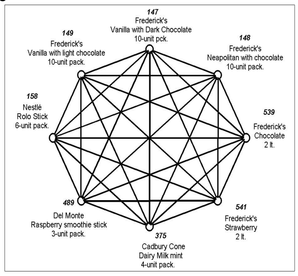
Cut-cliques applied to Market basket networks

Maximum cardinality clique

G **has 691 nodes and 1181 edges**

prod id number	# external links
147	4
148	13
149	3
375	9
489	16
518	0
539	0
541	0

45 links to 32 other products

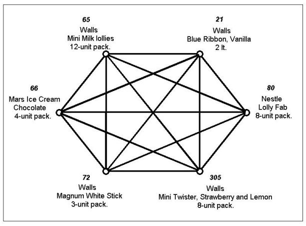


Cut-cliques applied to Market basket networks

Maximum cut-clique

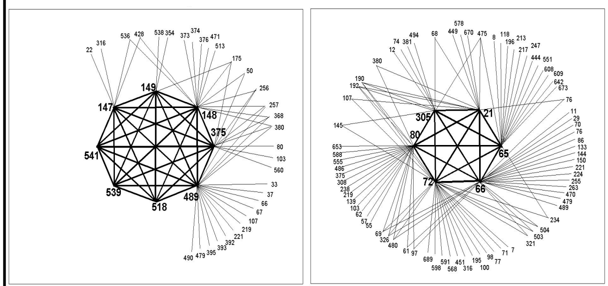
G **has 691 nodes and 1181 edges**

prod id number	# external links
21	10
65	14
66	23
72	23
80	21
305	9



100 links to 75 other products

Cut-cliques applied to Market basket networks



maximum cardinality clique

maximum cut-clique

Marketing Implications

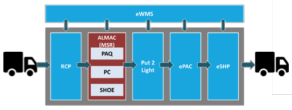
- The solution obtained is formed by the top-selling ice creams from leading brands: Walls, Nestlé and Mars.

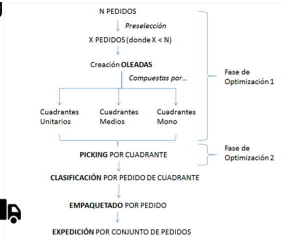
Marketing Implications

- The set of products constituting the maximum clique is not the clique with largest incidence to other products in the network.
- The householders buying the 6 products in the MCC are also strong potential buyers for the remaining products, especially those products involved in the 100 links.
- The MCC reveals interacting patterns from leading-sale products.

Warehouse Zara

- Optimization of the picking of the online orders
- This warehouse prepares 30,000 to 500,000 orders a day.





ILS 32

Warehouse Zara

- Input data
  - Online order to be prepared
  - Due date of the orders
  - Detail information of each SKU of the orders
  - Location of each SKU of the orders in the warehouse
  - Distance matrix of each location in the warehouse
- Decision Variables
  - Orders to be prepared in the next preparation (Oleada)
  - Orders associated with each group
  - Routing of the picking of each group

ILS 33

Warehouse Zara

- Constraints
  - Number of orders by group (size of the preparation area)
  - Number of groups by preparation phase (oleada)
  - Number of employees
  - Number of SKUs by group
  - Due date at packing area
  - Other special constraints…

ILS 34

Warehouse Zara

- Algorithm
  - Clustering, Assignment, Routing Algorithm (CARA)
  - Greedy Randomized Adaptive Search Procedure (GRASP)
  - Iterated Local Search (ILS)
- Results
  - Run the algorithm for the small instance with 500 orders, 2 area-stocks, 200 positions, and more than 400 items.
  - Be able ton prepare more 25% order per hour!

ILS 35

Hybrid ILS and Extensions

- Hydrid with other metaheuristics
- SimILS
- MathILS

Word cloud: Blum et al. (2011) Hybrid metaheuristics in combinatorial optimization: A survey. Applied Soft Computing 11:4135-4151

ILS 36

ILS… SimILS… MathILS…

- ILS represents one of the most efficient yet easy-to-implement frameworks for solving combinatorial optimization problems.
- **Easy to implement and adapt.**
- Most real-life problems are **complex** and filled with **uncertainty**.
- By integrating simulation inside the local search process, SimILS framework extends the virtues of ILS to stochastic COPs as well.
- The same by using exact methods, lower bounds etc…

ILS 37

UNIVERSITAT POMPEU FABRA

Hybrid ILS with other metaheuristics

- Use the ILS structure with other metaheuristics
- **Local Optimization** Phase
  - Tabu Search
  - VNS
  - Simulated Annealing
  - Variable Neighborhood Search
  - ….
- Perturbation Phase
  - Large neighborhood change

ILS 38

Distribution problem

- Extended Vehicle Routing Problem
  - Heterogeneous fleet (7 different truck capacities)
  - Time windows in the stores
  - Constraints of assigning some trucks to some stores.
  - Maximum driving hours
  - Multitrip for some vehicles
  - Sales constraints
- Minimize operative costs

ILS 39

Distribution problem

- GILS-VND Algorithm
  - GILS-VND, combina un Iterated Local Search (ILS), Greedy Randomized Adaptive Search Procedure (GRASP), y Variable Neighborhood Descent (VND).
  - ILS Structure
    - Initial solution: **GRASP**
    - Local improvement: **Variable Neighborhood Descend**
    - Perturbation: **Refine using random neighborhood**
  - Coelho V.N., Grasas A., Ramalhinho H., Coelho I.M., Souza M.J.F. (2016), An ILS-based Algorithm to Solve a Large-scale Real Heterogeneous Fleet VRP with Multi-trips and Docking Constraints, European Journal of Operational Research 250 (2): 367–376.

ILS 40

Distribution problem

- Computational results

Statistical results for the new set of instances: GILS-VND vs. Company.

		GILS-VND			
Instance	Company	Best	Average	Std. dev.	Gap (%)
K	32,472	30,507	30,692	70	−5.48
L	18,997	16,327	16,427	42	−13.53
M	20,266	18,017	18,146	43	−10.46
N	51,609	45,523	45,813	100	−11.23
O	45,764	40,846	40,995	66	−10.42

ILS 41

Distribution problem

- Results
  - Savings 10% daily with respect to the actual solutions.
    - A significant daily amount!
  - Savings of 2% compared with Prins' Algorithm with a simpler version of the problem.
  - Smaller number of vehicles need.
  - Better coordination with sales department.

ILS 42

SimILS

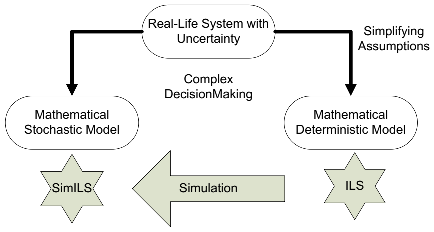
- Stochastic Combinatorial Optimization Problems
  - Uncertainty is present – Random Data
    - Example: Stochastic Demand in Vehicle Routing Problems
    - Stochastic processing times in scheduling
    - etc…
  - Strategic Problems
- Extends ILS to solve Stochastic Models…
- **SimILS**
  - Simulation + Iterated Local Search

ILS 43

UNIVERSITAT POMPEU FABRA

Stochastic Combinatorial Optimization Problems

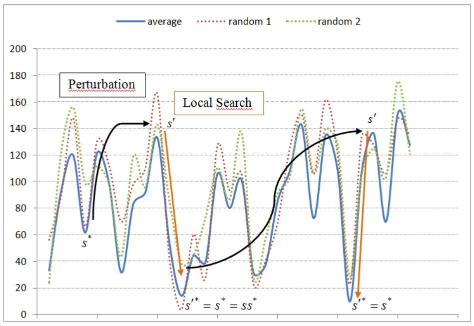
- SimILS



SimILS

```
Procedure SimILS
    s0 = GenerateInitialSolution
    s* = LocalSearch(s0)
    (s*, sf(s*), statistics) = Simulation(s*, long)
    Repeat
        s' = Perturbation(s*, history)
        s'* = LocalSearch(s')
        (s'*, sf(s'*), statistics) = Simulation(s'*, short)
        s* = AcceptanceCriterion(s*, s'*, history)
    Until termination condition met
    (s*, sf(s*), statistics) = Simulation(s*, long)
Return s*, sf(s*)
End
```


Exemplification of the SimILS for a COP with stochastic objective function



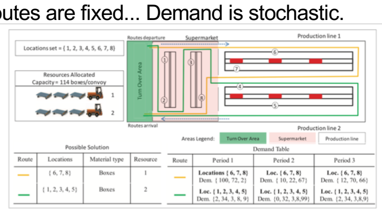
SimILS
Stochastic Constraints

```
Procedure SimILS
    s0 = GenerateInitialSolution (input data: average values)
     s* = LocalSearch(s0)
    (s*, sf(s*), service level) = Simulation(s*, long)
    Repeat
        s' = Perturbation(s*, history)
        s'* = LocalSearch(s')
        (s'*, sf(s'*), service level) = Simulation(s'*, short)
        s* = AcceptanceCriterion(s*, s'*, service level, history)
    Until termination condition met
    (s*, bsf(s*), service level) = Simulation(s*, long)
Return(s*; bsf(s*))
End
```


SEAT

Designing Internal supply routes

- Design a fix set of routes to supply components and materials to the production assembly line.
- Routes are fixed... Demand is stochastic.

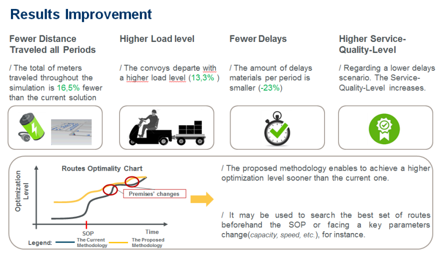


Joint work with Marcelus Fabri (UPF)

SEAT

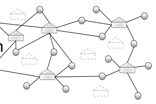
Designing Internal supply routes

- Working on a SimILS



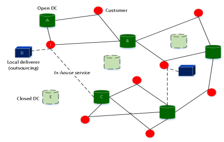
Supply Chain Design for ecommerce

- Supply Chain Design for ecommerce
  - Two-Stage Stochastic Programming Problem
- The goal is to find:
  - the subset of warehouses to be opened;
  - and determine the customer's assignment to the open warehouses
  - … such that all the demand is served at minimum total cost.
  - Demand is stochastic …
  - Each customer area must have 2 or 3 warehouses assigned as regular warehouses.



Supply Chain Design for ecommerce

- Two-Stage Optimization Problems
- The problems has two groups of variables interrelated among them …
  - Strategical decision variables (long term deterministic decision)
  - Operational decision variables (short term decisions), therefore stochastics at the moment of the decision process!



Supply Chain Design for ecommerce

- Solving this Two-Stage **Stochastic** Optimization Problem
- **Deterministic Equivalent Model (DEM)**
  - solved by CPLEX.
- **SimILS**
  - Simulation + Iterated Local Search
  - Use simulation to obtain the expected overall cost.
  - Local Search on open/close warehouses
  - Only *promising* solutions are tested in a stochastic environment.
- SimILS results compared favorably with **DEM**, with shorter running times.

Supply Chain Design for ecommerce

- Deterministic Equivalent Model (DEM)
  - Stochastic Programming

(SCFLPrp)

$$\min Z_{stoch} = \sum_{j=1}^{m} f_j y_j + \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{s} \pi_k c_{ij} d_{ik} x_{ijk}$$

s.t.

$\sum_{j=1}^{m} r_{ij} \le R,$ for $i = 1, \dots, n;$

$\sum_{j=1}^{m} x_{ijk} = 1,$ for $i = 1, \dots, n, \quad k = 1, \dots, s;$

$\sum_{i=1}^{n} d_{ik} x_{ijk} \le q_j y_j,$ for $j = 1, \dots, m, \quad k = 1, \dots, s;$

$x_{ijk} \le r_{ij},$ for $i = 1, \dots, n, \quad j = 1, \dots, m, \quad k = 1, \dots, s;$

$y_j \in \{0,1\}, r_{ij} \in \{0,1\}, x_{ijk} \ge 0$ for $j = 1, \dots, m; i = 1, \dots, n; k = 1, \dots, s$

Open/close Warehouse

Regular/not regular Warehouse

Demand assignment on scenario k

SimILS for SC Design Problem

```
Procedure SimILS
   s0=GenerateInitialSolution
   s*=LocalSearch(s0)          ← (a) open (b) close (c) open-close
   (s*,sf(s*),statistics)=Simulation(s*,long)
   Repeat
      s'=Perturbation(s*,statistics)   ← Destruction-Reconstruction process
      s'*=LocalSearch(s')
      (s'*,sf(s'*),statistics)=Simulation(s'*,short)
      s* =AcceptanceCriterion(s*, s'*,history)
   Until termination condition met
   (s*,sf(s*),statistics)=Simulation(s*,long)
Return s*,sf(s*)
End
```


Supply Chain Design for ecommerce

- Some results…
- cap11#,
  - 50 facilities
  - 50 customers
- Capa/b/c#,
  - 100 facilities
  - 1000 customers

	Stochastic Programming			SimILS		
Instance	Z_{sim}^{A}	t (sec)	gap Cplex (%)	Z_{sim}^{A}	t (sec)	gap (%)
cap111	879371.7	1103	0	883539.4	170	0.47
cap112	965382.2	3604	0.01	958047.4	303	-0.76
cap113	1047322.2	3603	0.04	1030653.5	294	-1.59
cap114	1160395.4	3604	0.06	1139246.8	50	-1.82
cap124	975397.6	3603	0.09	949375.9	65	-2.67
capa1	-			19244150.7	2263	
capa2	-			18458612.4	2370	
capa3	-			17827599.1	2934	
capa4	-			17162132.3	963	
capb1	-			13773560.0	809	
capb2	-			13378972.7	2645	
capb3	-			13238010.0	2086	
capb4	-			13084859.8	2380	
capc1	-			11704267.9	1363	
capc2	-			11571933.3	1812	
capc3	-			11540950.9	2850	
capc4	-			11535842.8	1928	

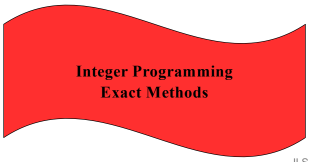
UNIVERSITAT POMPEU FABRA

upf. Universitat Pompeu Fabra Barcelona

Solution Methods

- Good solutions for complex and large-scale problems
- Short running times
- Easily adapted

- Mathematical proved optimal solutions
- Important information on the characteristics and properties of the problem.



ILS 56

upf. Universitat Pompeu Fabra Barcelona

Math - Iterated Local Search

- Get an initial solution x;
  - * Heuristic method or a random solution.
  - * local optimization method
- For a certain number of iterations:
  - **Perturbation Step**
    - * **Uses a exact method to solve a subproblem or a relaxation of the problem.**
  - Small-steps
    - * local optimization method, initial solution x'; final solution x''.
  - Perform an accept/reject test
    - * accept all solutions, accept with a certain probability or accept only if it is a better solution.
    - * If x'' is accepted, then x = x''.
- Return the best solution found.

ILS 57

upf. Universitat Pompeu Fabra Barcelona

MathILS

- Maybe the first application…
  - Use an exact algorithm to solve a sub problem within a **Iterated Local Search** heuristic for the Job-Shop Scheduling Problem
  - **Solving to optimality the one-machine scheduling problem with due dates and delivery times using the Carlier Algorithm.**
    - * **Lourenço H.R.** (1995), Job-Shop Scheduling: computational study of local search and large-step optimization methods. European Journal of Operational Research **83**(2): 347-364. ISSN 0377-2217.
    - * PhD Thesis, Lourenco H.R. (1993)

ILS 58

upf. Universitat Pompeu Fabra Barcelona

Example of Applications

- Real Applications
  - Maybe the best set of problems to apply Matheuristics methods…
  - Why?
    - * Complex problems with a large number of constraints.
    - * Sometimes difficult to model…
    - * But, a simplification of the problem is frequently a well-studied optimization problem.
  - Apply **metaheuristics for the real general problem**…
  - And **exact methods for the well-known relaxation problem**.

ILS 59

upf. Universitat Pompeu Fabra Barcelona

Metaheuristics

- Which is the best metaheuristic?
  - Begin with a simple method and then turn, if necessary, to a more complicated one or refine the first implementation
  - Small number of parameters
  - Evaluate its performance by:
    - * Accuracy
    - * Speed
    - * Simplicity
    - * Flexibility

ILS 60

upf. Universitat Pompeu Fabra Barcelona

Conclusions

- Iterated Local Search
  - Simple
  - Easy to implement
  - Robust
  - Highly effective
  - Modularity
- Start simple and add complexity if needed!
- The success of ILS lies in the biased sampling of the set of local optimal.
- More than 6000 publications in google scholar.

Do you want to try to implement an ILS?

ILS 61

Iterated Local Search

- Main References
  - ***Lourenço H.R.***, Martin O. and Stützle T. (2010), Iterated Local Search: Framework and Applications. In Handbook of Metaheuristics, 2nd. Edition. Vol.146. M. Gendreau and J.Y. Potvin (eds.), Kluwer Academic Publishers, International Series in Operations Research & Management Science, pp. 363-397.
  - ***Lourenço H.R.***, Martin O. and Stützle T. (2003), Iterated Local Search. In Handbook of Metaheuristics, F. Glover and G. Kochenberger, (eds.), Kluwer Academic Publishers, pp. 321-353.
  - Grasas A., Juan, A.A. and ***Lourenço H.R.*** (2014), SimILS: A Simulation-based extension of the Iterated Local Search metaheuristic for Stochastic Combinatorial Optimization, Journal of Simulation doi:10.1057/jos.2014.25

ILS 62