Metaheuristics for transportation problems

Abdelghani BEKRAR
Associate professor
abdelghani.bekrar@univ-valenciennes.fr

Agenda

- Introduction
- Optimization problems in Transportation
- Complexity
- Solving Methods
  - Exact methods
  - Heuristics
  - Metaheuristics
- Classification of metaheuristics
- Local search method
- Simulated annealing

Agenda

- Introduction
- Optimization problems in Transportation
- Complexity
- Solving Methods
- Exact methods
- Heuristics
- Metaheuristics
- Classification of metaheuristics
- Local search method
- Simulated annealing

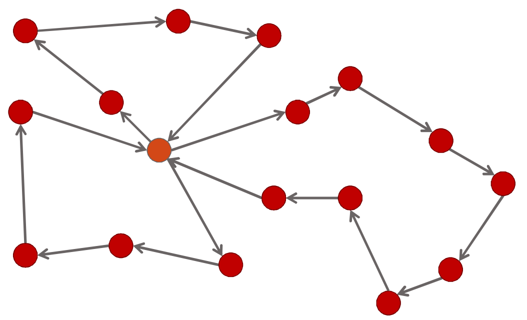
Introduction: Optimization Problem

- A combinatorial optimization problem (COP) belongs to the family of optimization problems where the set of feasible solutions is discrete or can be reduced to a discrete set,
- The objective is to find the **best** possible solution.

Vehicle Routing Problem

- Given a list of customers, distances between them and a set of vehicles, find tours that minimize the total length of the tours, such that one vehicle visits each location
- Formulated in 1959
- Typically, one has to serve a scattered set of customers from a single central depot, such that each vehicle has a limited capacity

VRP



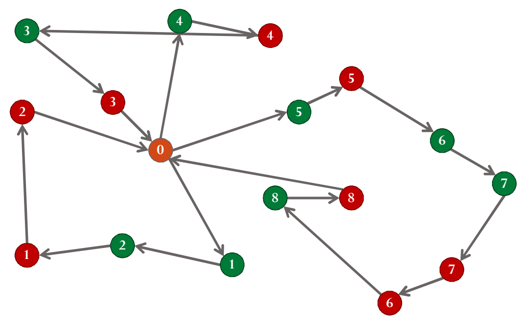
Vehicle Routing Problem Variants

- VRP with time windows (VRPTW)
- Fleet size and mix VRP (FSMVRP)
- Open VRP (OVRP)
- Multi-depot VRP (MDVRP)
- Periodic VRP (PVRP)
- VRP with backhauls (VRPB)
- Pickup and delivery problem (PDP)
- Dynamic VRP (DVRP)
- VRP with stochastic demands (VRPSD)

Pickup and Delivery Problem

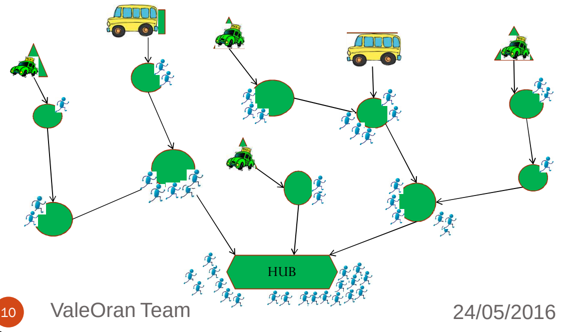
- Each task consists of two parts
  - Pickup
  - Delivery
- VRP (and MDVRP) a special case of PDP
- Can be combined with other aspects
  - Time windows, capacity, fleet size and mix, …
- Real-life examples include oil transportation, school buses, courier services, …

PDP



Optimisation in Transport

An Improved Tabu Search (ITS) heuristic algorithm for the Coach Trip with Shuttle Service Problem, VeRoLog 2015,



Combinatorial optimization problem

Definition

- *An instance of COP is a pair (S, f), where S is the set of feasible solutions that each have a cost or profit f. f is a defined function from S to R.*
- *The objective is to find the optimal solution, i.e.: $x^* \in S$ such that $f(x^*) \leq f(x)$; $\forall x \in S$ (in case of minimization problem).*
- *$f^* = f(x^*)$ is the optimal cost or profit and $S^* = \{ x \in S \mid f(x) = f^* \}$ is the set of optimal solutions.*

Exemples of COP

- Knapsack problem
- Supply chain Optimization
  - Production planning
  - Procurement,
  - Lot sizing
- Vehicle routing problem (VRP),
- Scheduling or Sequencing problems in shop floor,
- Loading and bin packing problems

Example of COP

Capacitated Lot-Sizing Problem (CLSP)

- Demands for *N* products known on a planning horizon of length T periods.
- The production process requires a processing time and preparation time (setup) on a production unit (machine) of limited capacity.

- **The costs:**
  - Processing, setup
  - Handling (stoking).
  - Penalities on backlogs

- The **objective** is to determine the periods of production and the production quantities in these periods to minimize the total cost while meeting all demands. (when and how)

Exemple de POC

- On considère une seule ressource
- Les coûts considérés pour chaque produit *i* et chaque période *t* sont :
  - le coût de production p_{it},
  - le coût de préparation s_{it} qui est généré chaque fois que la production a lieu,
  - et le coût de stockage h_{it}.

Why using metaheuristcs?

Agenda

- Introduction
- Optimization problems in Transportation
- Complexity
- Solving Methods
- Exact methods
- Heuristics
- Metaheuristics
- Classification of metaheuristics
- Local search method
- Simulated annealing

Complexity of optimization problems

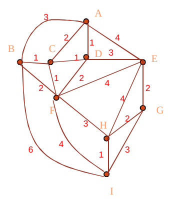
- Some problems are Easy to solve and but other are difficult,
- A Problem => solving method (algorithm)
- Algorithmic complexity vs Complexity of problems
- reference : **M. R. Garey et D. S. Johnson.** ***Computers and intractability, a guide to the theory of NP-completeness.*** **Freeman, New York, 1979**.

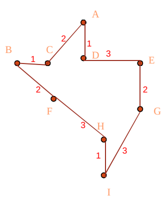
Complexity

The Travelling Salesmen Problem (TSP)

- A set of cities (n)
- A traveler who travels all cities and returns to the starting city,
- He visits each city once
- Find the shortest path
- **A simple statement,**

TSP, example





Complexity

Naive solution :

- List all possible paths,
- Take the shortest one,
- At the first city, we have n-1 possibilities,
- At the second we have n-2,…
- 1*(n-1)*(n-2)*….= n!
- For $n = 24$, $24! \approx 3 \times 10^{23}$.
- If we take a *nanosecond* to calculate the cost of each circuit ➔ we will need 3×10^{14} secondes, around 10 million years.

Complexity

- We assume that each basic operation takes constant time,
- The efficiency of an algorithm is the number of basic operations that it performs,
- It depends on the size of the input data and not from the computer
- The number of operations in the TSP algorithm is (n-1)!
- We note it: **O(n!),**
- The algorithm is said polynomial or in polynomial time if the number of operations is a polynomial function,
- When the function is not polynomial, we said it is exponential even if the function is not realy.

Complexité algorithmique

Complexity	Ref Computer	Computer 100 times faster	Ordinateur 1000 times faster
N	N1	100*N1	1000*N1
N^2	N2	10*N2	31,6*N2
N^3	N3	4,64*N3	10*N3
2^N	N4	N4+6,64	N4+9,97
3^N	N5	N5+4,19	N5+6,29

Computers and intractability, a guide to the theory of NP-completeness(Garey & Johnson)

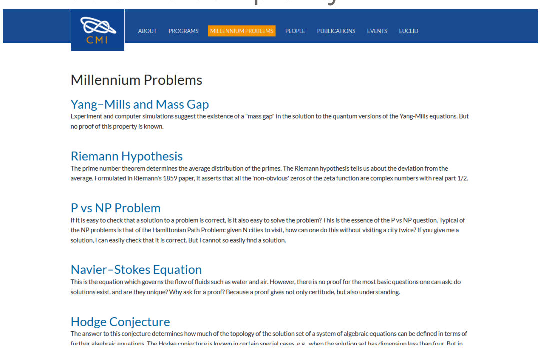
Problem's Complexity

- Related to the nature of the problem
- Is independent of used methods or algorithms,

Complexity Classes:

- The complexity theory mainly studies the problems of decision,
- This is to answer the question with "Yes" or "No" to the question "Is the problem has an optimal solution"
- **Decision problem**: each combinatorial optimization problem can be transformed into a decision problem. The optimization problem of the function f (x) is transformed into a decision problem: Is there a solution x such that f (x) <k, for a given k.
- For the TSP, the associated decision problem is the existence of a Hamiltonian path (CH) with a lower cost than an integer k.
- *CH in a graph is a path that contains all the summits and passes exactly once through each vertex*

Problem's Complexity


CMI

ABOUT PROGRAMS MILLENNIUM PROBLEMS PEOPLE PUBLICATIONS EVENTS EUCLID

Millennium Problems

Yang–Mills and Mass Gap

Experiment and computer simulations suggest the existence of a "mass gap" in the solution to the quantum versions of the Yang-Mills equations. But no proof of this property is known.

Riemann Hypothesis

The prime number theorem determines the average distribution of the primes. The Riemann hypothesis tells us about the deviation from the average. Formulated in Riemann's 1859 paper, it asserts that all the 'non-obvious' zeros of the zeta function are complex numbers with real part 1/2.

P vs NP Problem

If it is easy to check that a solution to a problem is correct, is it also easy to solve the problem? This is the essence of the P vs NP question. Typical of the NP problems is that of the Hamiltonian Path Problem: given N cities to visit, how can one do this without visiting a city twice? If you give me a solution, I can easily check that it is correct. But I cannot so easily find a solution.

Navier–Stokes Equation

This is the equation which governs the flow of fluids such as water and air. However, there is no proof for the most basic questions one can ask: do solutions exist, and are they unique? Why ask for a proof? Because a proof gives not only certitude, but also understanding.

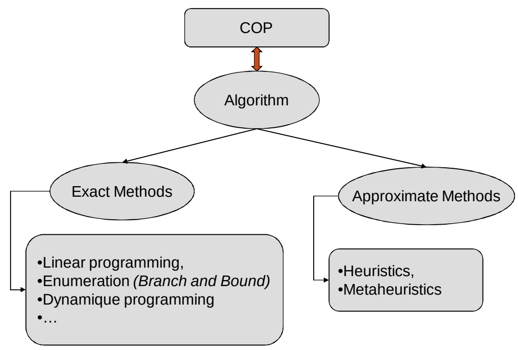
Hodge Conjecture

The answer to this conjecture determines how much of the topology of the solution set of a system of algebraic equations can be defined in terms of further algebraic equations. The Hodge conjecture is known in certain special cases, e.g., when the solution set has dimension less than four. But in

Agenda

- Introduction
- Optimization problems in Transportation
- Complexity
- **Solving Methods**
- Exact methods
- Heuristics
- Metaheuristics
- Classification of metaheuristics
- Local search method
- Simulated annealing

Solving methods of a COP



Solving methods of a COP

- **Exact Methods:** are methods that seek an optimal solution for a COP.

- **Advantage** : optimale solution => satisfactory, with minimum cost and maximum benefit,

- **Disadvantage** :
    - Time consuming,
    - For some problems (Difficult) it is **impossible** to determine an optimal solution.
    - Limited by the size of the problem

- **Approximate Methods** : constructif, based on experience,
- **Advantage** : fast and efficient,
- **Disadvantage** :
    - No guarantee of the solution optimality,
    - Complex and different parameters for each problem instance.

Agenda

- Introduction
- Optimization problems in Transportation
- Complexity
- Solving Methods
- **Exact methods**
- Heuristics
- Metaheuristics
- Classification of metaheuristics
- Local search method
- Simulated annealing

Linear Programming

- Is to determine the *n* variables (decisions) that optimizes (maximize or minimize) the objective function.
- Min/Max f(x) = amount of cost(or amount of profit)
- Respect the constraints (capacity, order, …)
- When variables are integer, we talk about Integer Linear Programming (ILP),
- When some variable are continue (in R), we talk about Mixed Integer Programming (MIP),
- Two steps:
  - Model,
  - Solve: simplex, Solver(GLPK, Coin-OR, IBM-Cplex, Excel, MSF,…)

Example: LP for CLSP

Parametres and notations

- For each periode t et and item i
- Setup cost s_{it}
- Handling cost h_{it}
- Processing cost p_{it}
- Customer demand D_{it}
- Processing time σ_{it}
- Setup time τ_{it}
- ressource capacity C_t

Variables

- x_{it} : Quantity to make for item i at the periode t.
- y_{it} =1 if we produce i in the periode t (i.e. si $x_{it} > 0$).
  - 0 sinon
- I_{it} : stock level in period t for item i.

PL pour CLSP

$$Min \sum_{i=1}^{N} \sum_{t=1}^{T} (s_{it} Y_{it} + p_{it} X_{it} + h_{it} I_{it})$$

$$sc:$$

$$I_{i(t-1)} + X_{it} = D_{it} + I_{it} \qquad \forall i,t$$

$$\sum_{i=1}^{N} (\sigma_{it} X_{it} + \tau_{it} Y_{it}) \le C_t \qquad \forall t$$

$$X_{it} \le (\sum_{l=t}^{T} D_{il}) Y_{it} \qquad \forall i,t$$

$$Y_{it} \in \{0,1\} \qquad \forall i,t$$

$$X_{it} \ge 0, I_{it} \ge 0 \qquad \forall i,t$$

Agenda

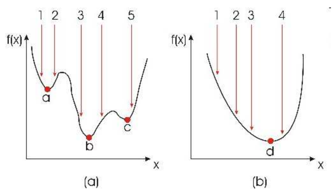
- Introduction
- Optimization problems in Transportation
- Complexity
- Solving Methods
- Exact methods
- **Heuristics**
- Metaheuristics
- Classification of metaheuristics
- Local search method
- Simulated annealing

Heuristics

- **Heuristic** from Greek *eurisko*, « find or discover », i.e « the art of inventing, make discoveries »
- It is an approximatif algorithme that provide polynomial time a feasible solution, not necessary optimal, for COP,
- In general, it is designed for a specific proble, based on its particular structure,
- Can be based on experience,
- It is generally Iterative
- To be effective in terms of computation time, it should be simple and if possible in one pass,
- To prove the effectiveness of a heuristic, we try to check that it guarantees some performance,
- **Risk**: trapped in a local optimum

Heuristics

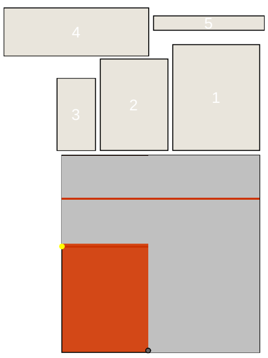
- Local Optimum ,



Heuristics

The SHF heuristic for the *2D Bin Packing problem* :

- Sort items by increasing heights
- Placing in layers
- Placement in available rectangles,
- Complexity of O (n2)
- Respect for guillotine constraints
- Fall rate 11%



Agenda

- Introduction
- Optimization problems in Transportation
- Complexity
- Solving Methods
- Exact methods
- Heuristics
- **Metaheuristics**
- Classification of metaheuristics
- Local search method
- Simulated annealing

What is a Metaheuristic?

- Meta-heuristic is a top-level strategy that guides an underlying heuristic solving a given problem. Stutzle (1999)
- it "refers to a master strategy that guides and modifies other heuristics to produce solutions beyond those that are normally generated in a quest for local optimality." Glover.
- "A metaheuristic is an iterative generation process which guides a subordinate heuristic by combining intelligently different concepts for exploring and exploiting the search spaces using learning strategies to structure information in order to find efficiently near-optimal solutions." Osman and J.P. Kelly (1996),
- "A metaheuristic is a set of concepts that can be used to define heuristic methods that can be applied to a wide set of different problems. In other words, a metaheuristic can be seen as a general algorithmic framework which can be applied to different optimization problems with relatively few modifications to make them adapted to a specific problem." Metaheuristics Network Website (2000).

Common features

- Starting from an initial solution and look better,
- Using the concept of 'neighborhood'
- Are strategies that guide the search process,
- Effectively explore the search space to find a solution close to the optimum,
- Escape the local minima
- Inspired by the natural experiments (tabu, genetic algorithms, ant colonies, Swarm) or physical (simulated annealing, electromagnetism)

Agenda

- Introduction
- Optimization problems in Transportation
- Complexity
- Solving Methods
- Exact methods
- Heuristics
- Metaheuristics
- **Classification of metaheuristics**
- Local search method
- Simulated annealing

Classification of metaheuristics

Different classifications was proposed

- ***The origine of the algorithme :***
  - Inspired from nature or not (bio-inspired, GA, SA, …)
  - Some recent algorithms do not fit into this classification
- ***The number of used solutions:***
  - Population based Algorithms (GA, ANT)
  - Sigle solution algorithms (taboo search, ILS, VNS)
- ***Neighborhood:***
  - Most of metaheuristics use a single neighborhood
  - Some metaheuristics, such as variable neighborhood search (Variable Neighborhood Search, VNS), use several neighborhoods,
  - Diversification,
- ***Memory Usage:***
  - Metaheuristics use the history to move forward in exploration
  - Others use memory more "long" to accumulate a summary of search parameters.

Agenda

- Introduction
- Optimization problems in Transportation
- Complexity
- Solving Methods
- Exact methods
- Heuristics
- Metaheuristics
- Classification of metaheuristics
- **Local search method**
- Simulated annealing

Local search method

- Local search is a metaheuristic family based on the concept of neighborhood.
- Local search includes descent techniques, or the gradual improvement of a solution.
- For a minimization function, we calculate a decreasing costs of solutions.

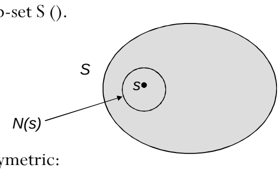
Local search method

Methodology

- To solve an optimization problem with local search approach, we define:
  - Search space S
  - Cost function f : S -> R
  - Define a neighborhood
  - Choose the mechanism to visit different configurations

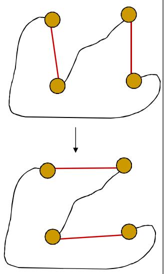
Local search method

- Set *S* is the space of the solution associated with a COP of minimization P and *s* is the current solution,
- f is the objective function that calculate for each solution s its cost; f : s ∈ S ➔f(s)
- *The neighborhood N(s)* of s is sub-set S ().



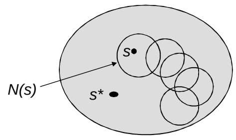
- *The neighborhood* N is usually symetric:
  s' belong to *N(s)* if *s* belong to *N(s')*.

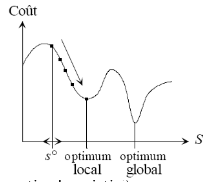
Local search method



- Algorithm
  - Compute an initial solution
  - **loop**
    - explore $N(s)$
    - If $f(s') \leq f(s)$ **then** s := s'
  - **While** $f(s') \leq f(s)$, $\forall\, s' \in N(s)$
- $N(s)$ is generally defined from simple transformations or *moves*
- Example TSP: Replace two non-adjacent edges (figure)

Local search method





- Random initial solution or good constructive heuristic?
- Can fall into a local optimum f (s) ≤ f (S), for all *s'* in *N (s)*.
- Exploration of N(s) must be fast: $O(n^2)$ or $O(n^3)$
- Several ways of choosing the neighbor N (s):
  - *first improvement*
  - *best improvement*

Local search method

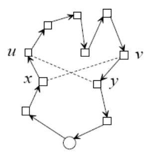
Example TSP

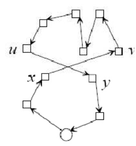
- Search space
  - A possible solution is any hamiltonian cycle (route).
- Minimization function(objective)
  - For each solution s, f(s) is the length of the route s (amount of edges length).
- *Neighborhood*
  - A possible *neighborhood* 2-opt.
  - A 2-opt move consist in:
    - Replace two non-adjacent edges (a, b) and (c, d) with
    - Two edges(a, c) and (b, d).

Local search method

K-OPT neighborhood

- Replace k edges with k other edges .
- Cost $O(n^k)$, k = 2 ou 3 in practice.
- Example of 2-OPT with (x,u) and (v,y):





- Cost variation: $\Delta = d(x,v) + d(u,y) - d(x,u) - d(v,y)$

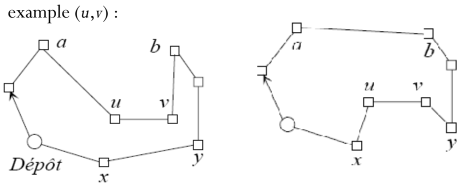
Local search method

- Algorithm 2-OPT
- **repeat**
  - Δmin := ∞
  - **for each** pair (u,v) of vertex while u is before v **do**
    - Let x *the predecessor* of u et y the sucessor of v
    - $\Delta := d(x,v) + d(u,y) - d(x,u) - d(v,y)$
    - **if** $\Delta \leq \Delta$min **then**
      - Δmin := Δ; *bestu* := u; *bestv* := v
    - **endif**
  - **endfor**
  - **if** Δmin < 0 **then**
    - Reverse in the route T the sub-chaine of *bestu* to *bestv*
  - **endif**
- **until Stop (if no improvement)**

Local search method

Neighborhood OR-OPT

- Move a chaine of 1, 2 or 3 vertex,
- example (u,v) :



- $\Delta = D(a,b)+D(y,v)+D(u,x)-D(a,u)-D(v,b)-D(y,x)$
- Exploration with $O(n^2)$

Exercice

- We consider the following ***knapsac problem:***

#object	1	2	3	4	5	6
Value	40	40	30	18	20	1
Volume	21	22	17	13	16	2

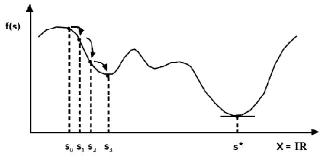
- We have 6 objects to put in the sac of capacity (volume) 55 aiming to maximize the profit.
- Apply the local search to find the best solution

Agenda

- Introduction
- Optimization problems in Transportation
- Complexity
- Solving Methods
- Exact methods
- Heuristics
- Metaheuristics
- Classification of metaheuristics
- **Local search method**
- Simulated annealing

Simulated annealing

- The disavantage of the local search



- How to escape the local optimum?
  - Simulated annealing(Kirkpatrick & al., 1983)
  - Acceptance with threshold (Dueck & Scheuer, 1990)
  - Tbou search (Glover, 1986 ; Hansen, 1986)

SA: History

- Developed by different researchers in the 80s,,
- Proposed first by S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi de la société IBM en 1982. [Vecchi, M. and Kirkpatrick, S. (1983). Global wiring by simulated annealing. *IEEE Trans. on C.A.D.*, CAD-2(4):215–222].
- Similair work was presented in 1985 by V. Cerny,
- Inspired by a physical phenomenon,
- A method that avoids the local optimum
- Since, it has been proven for some COPs and limitations facing other,
- This is the first metaheuristic that has been proposed.
- Easy to implement

Analogy: COP vs physical System

- Annealing is a technique used by physicists to bring a physical system in a state of low energy,
- They use the parameter "Temperature"
- Then lower heat the material in a controlled manner the temperature the temperature to reach a state of low energy,
- How to apply this setting "Temperature" the COP

Système physique	Problème d'optimisation
Free energy (low)	Objectif function
particules cordiantes	Problem parameters
Find the state with low temperature	Fin the good (best) Configuration

SA: main idea

Thermodynamic equilibrium

- probability (E,T) = $e^{-E/KbT}$
  - E : energy, T : temperature, K_b : Boltzmann constant

Passing from one state to another

- Calculate ΔE : probability (ΔE,T)
- If $\Delta E \leq 0$: we improve the objectif
  - Accept the transformation
- If $\Delta E > 0$: on "dégrade" la fonction objectif
  - on accepte la transformation avec une probabilité (ΔE,T)

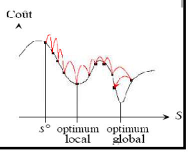
Exponentiel function

- IfT is large, the probability is near 1
  - We accept the degradation
- IfT is small, near 0
  - The most moves that increase the energy are rejected

SA: main idea

In optimization :

- We accept the new neighbor if the cost decreases.
- It is also accepted if the cost increases (rebound) but with a probability,
- Probability of accepting a rebound : $e^{-\Delta/T}$, ($\Delta > 0$ cost variation)
- The parameter T is called temperature.
- The probability tends to 0 during iterations.
- The probability decreases with the increase of the cost.



SA algorithm

We define:

- An initial solution s_0,
- An initial temperature T_0,
- The Neighborhood $N(s)$
- A *Cooling Schedule* function: $CS(T)$,
- Stop criteria

SA algorithm, simple

- Set the initial solution s
- T := initial temperature T_0
- **repeat**
  - Take randomly a neighbor s' in $N(s)$
  - $\Delta = f(s)-f(s')$
  - random = $e^{-\Delta/T}$
  - Generate randomly x uniformly in [0,1[
  - **if** $\Delta < 0$ **or** random>x **then**
    - s := s'
  - T := CS(T)
- **until** (stop criteria)

SA algorithm

Some recommendations :

❑ T should decrease slowly: *Cooling Schedule CS(T)*

❑ example $CS(T) := k*T$, $k = 0.999$, or (0<k<1)

❑ Stop criteria: $T < \varepsilon$, final temperature

❑ The best solution is the last

SA algorithm, complete

- Set the initial solution s
- T := initial temperature T_0
- ni := 0; //nbr of iterations
- ne := 0; // nbr of iterations without improvement
- bs := s; //best solution
- ***repeat***
  - ***for*** (niter, niter<niter_palier, niter++)
    - tirer au sort s' dans *N(s)*
    - $\Delta = f(s)-f(s')$
    - random = $e^{-\Delta/T}$
    - Generate *x* uniformly in [0,1[
    - **if** Δ < 0 **or** random>x **then**
      - s := s'
    - **if** f(s) < f(bs) **then** bs := s; ne := 0
    - **else** ne = ne+1
    - ni = ni +1
  - T := CS(T)
- ***until*** (T < ε) ou (ni = nimax) ou (ne >= nemax)

Example : Solving the Sudoko with SA

- A square grid of 81 squares divided into 9 squares 3 squares of side
- Partially filled with numbers 1 to 9
- The goal is to complete the filling of the grid with numbers from 1 to 9:
  - no number appears twice in the same line,
  - in the same column or
  - in the same sub-square.

	5	9	4	1				2
3		7	2	8		4		6
2	4		7	9	6	5		
9	6	5		7	2	8	4	3
8	7		9		4	1		5
4	3	1	5	6		2	7	
	2	6	3	4	1	9	5	8
5	8	4				3	1	7
		3	8	5	7		2	4

Example : k-coloriage

- Let the graph G = (V, E), V is the set of vertex ,
- E is the of pair of edges,
- Two vertexes v_1 and v_2 are adjacent if they are linked the edge ($\{v_1, v_2\} \in E$),
- The k-coloriage of G is the function

c : S ➔ [1, k]

Such that c(i) ≠c(j) when i and j are adjacent.

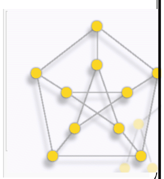
- *c(v)* is the color of the vertex *v,*
- The *k-coloriage* problem is *NP-Hard*

for k >2

- The objective function : for each configuration *S*,

f(S) is the number of edges « violated » in *S*.

The objective is to minimize this number.



Example : Solving the Sudoko with SA

- Sudoku is equivalent to complete a 9-coloring of a graph G = (V, E)
- Vertices are the grid boxes
- two vertices are connected by an edge if they belong to the same line in the same column or the same sub-square.
- application $f : V \longrightarrow \{1, 9\}$, telle que $f(v) \neq f(w)$ if {v, w} ∈ *E*.
- Can we complete a Sudoku? Is *NP*-complet

Exemple : le RS et le Sudoko

- The set of solutions S is all square grids 9 side whose cells contain a number from 1 to 9,
- N = |S| = 9^{81-m}, m is the *fixed cells*,
- Let s ∈ S , s_{ij} : the contents of the cell of row *i* and column *j*.
- $f_{ij}(s)$ is the number of occurrences of the digit s_{ij} in the row, column and square,
- *The objective function :*

$$f(s) = \sum_{i=1}^{9} \sum_{j=1}^{9} f_{ij}(s)$$

Example : Solving the Sudoko with SA

- Neighborhood :
  - Two grids are adjacent to each other if one can be obtained from the other by changing the number and content in one box (not fixed of course) ⇔ 1-opt.

- Start the algorithm with height température,
- Decrease the temperature using the following CS. =>

with τ small real positive,

$$CS(T): T_{k+1} = \frac{T_k}{1 + \frac{\log(1+\tau)}{ep+1}}$$

Example : Solving the Sudoko with SA

Algorithm RS_SUDOKUS

- $\tau = 0.1$, $e_p = 1620/2$, $T \leftarrow e_p$.
- $f \leftarrow f(s)$.
- **While** $T \geq 0.00273852$ **Do**
  - **for** $k = 1$ to 81 do //Palier
    - Choose *i* and *j* uniformly in {1, 9}
    - **while** « cell *ij is fixed* » do $t \leftarrow s_{ij}$ et $c_1 \leftarrow f_{ij}(s)$.
    - Choose s_{ij} uniformly in {1, 9}
    - **while** $s_{ij} = t$ **do** $c_2 \leftarrow f_{ij}(s)$ et $c' \leftarrow f - c_1 + c_2$
    - Choose u uniformly in [0, 1].
    - **if** $u \leq e^{-(c'-c)/T}$ ***then*** $c \leftarrow c'$, // (accept).
    - **else** Do $s_{ij} \leftarrow t$, // (rejet).
    - If $c = 0$ then save S and Stop.
  - **endfor //**
  - $T \leftarrow CS(T)$ ·
- end.
- Results : For a sudokus with 23 to 26 fixed cells, we need 2 to 7 trials.

Advantages and disadvantages of simulated annealing

Advantages

- usually leads to good solutions,
- Under certain conditions, simulated annealing converges in probability to a global optimum
- A generic method: applicable to many problems and easy to implement,
- Flexible: possibility to add constraints to solve the problem,

Disadvantages

- Many setting: find the "perfect "
- significant execution time-dependent of the setting

some references

- *Johann Dréo, Alain Pétrowski, Patrick Siarry, Eric Taillard,* **Metaheuristics for Hard Optimization: Methods and Case Studies,** Eyrolles - 09/2003 17 x 23 - 356 pages ISBN : 2-212-11368-4
- **Simulated annealing: Theory and applications**, P.J.M Laarhoven & E. Aarts, 1987.
- C.R., Reeves, ***Modern Heuristic Techniques for Combinatorial Problems***, Mc Graw-Hill, Advances topics in computer science. 1995

Exercice

- We consider the following TSP.
- Propose an initial solution,
- What can be the neighborhood
- Apply some iteration of the SA.