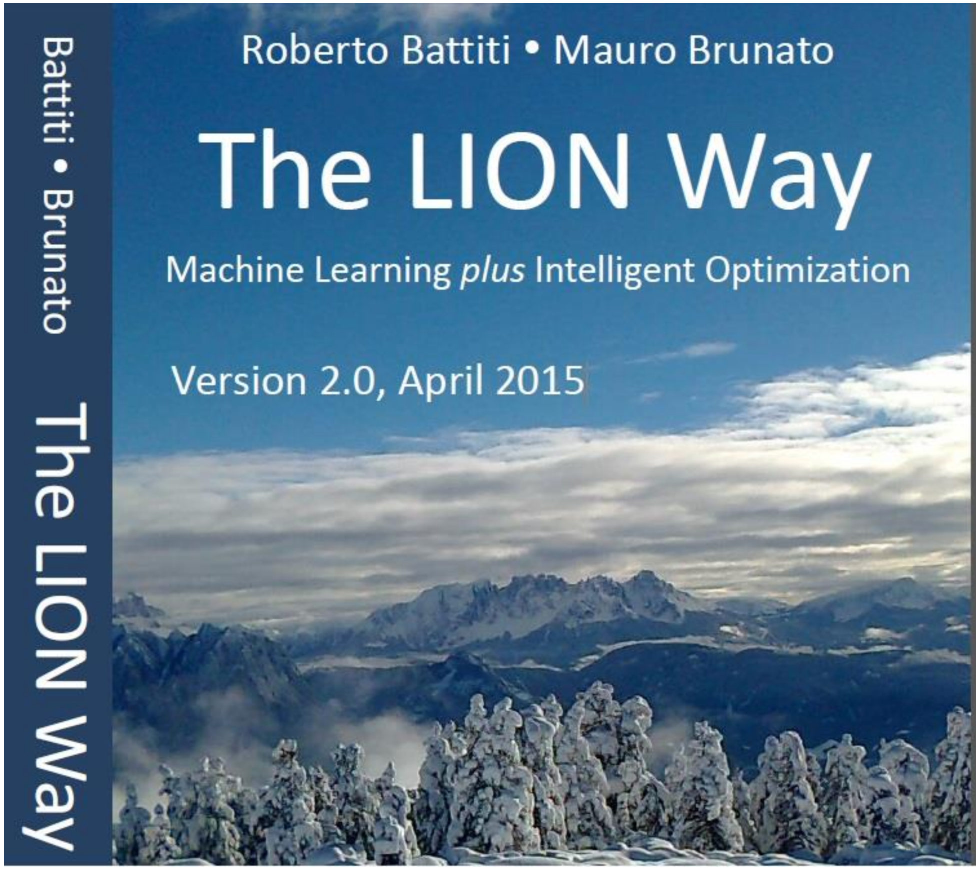


ROBERTO BATTITI, MAURO BRUNATO.
The LION Way: Machine Learning plus *Intelligent Optimization*.
LIONlab, University of Trento, Italy,
Apr 2015

http://intelligent-optimization.org/LIONbook

Local Search and Reactive Search Optimization (RSO)

Everybody carries on his shoulders the responsibility of his choices. It is a nice weight. (Romano Battiti)

Brute force is not the solution

- Let's assume that one has to find the minimum of a discrete (combinatorial) optimization problem (for example, think about the *travelling salesman* problem)
- Evaluating all possible combinations of inputs can be computationally impossible
- One needs to resort to clever techniques to solve these problems

Local search based on perturbations

- starting from an initial tentative solution
- try to improve it through repeated small changes
- stop when no improving local change exists

(local optimum, or locally optimal point)

Local search optimization: notation

- χ is the search space
- $X^{(t)}$ is the current solution at iteration t.
- $N(X^{(t)})$ is the neighborhood of point $X^{(t)}$, obtained by applying a set of basic moves $\mu_0, \ldots, \mu_M$ to the current configuration

$$N(X^{(t)}) = \{X \in \mathcal{X} \text{ such that } X = \mu_i(X^{(t)}), i = 0, \ldots, M\}.$$

Local search optimization

- Local search starts from an admissible configuration $X^{(0)}$ and builds a trajectory $X^{(0)},\ldots,X^{(t+1)}$.
- The successor of the current point is constructed as follows

$$
\begin{aligned}
Y &\leftarrow \textsc{Improving-Neighbor}(\ N(X^{(t)})\ ) \\
X^{(t+1)} &= \begin{cases} Y & \text{if } f(Y) < f(X^{(t)}) \\ X^{(t)} & \text{otherwise (search stops).} \end{cases}
\end{aligned}
$$

- IMPROVING -NEIGHBOR returns an improving element in the neighborhood

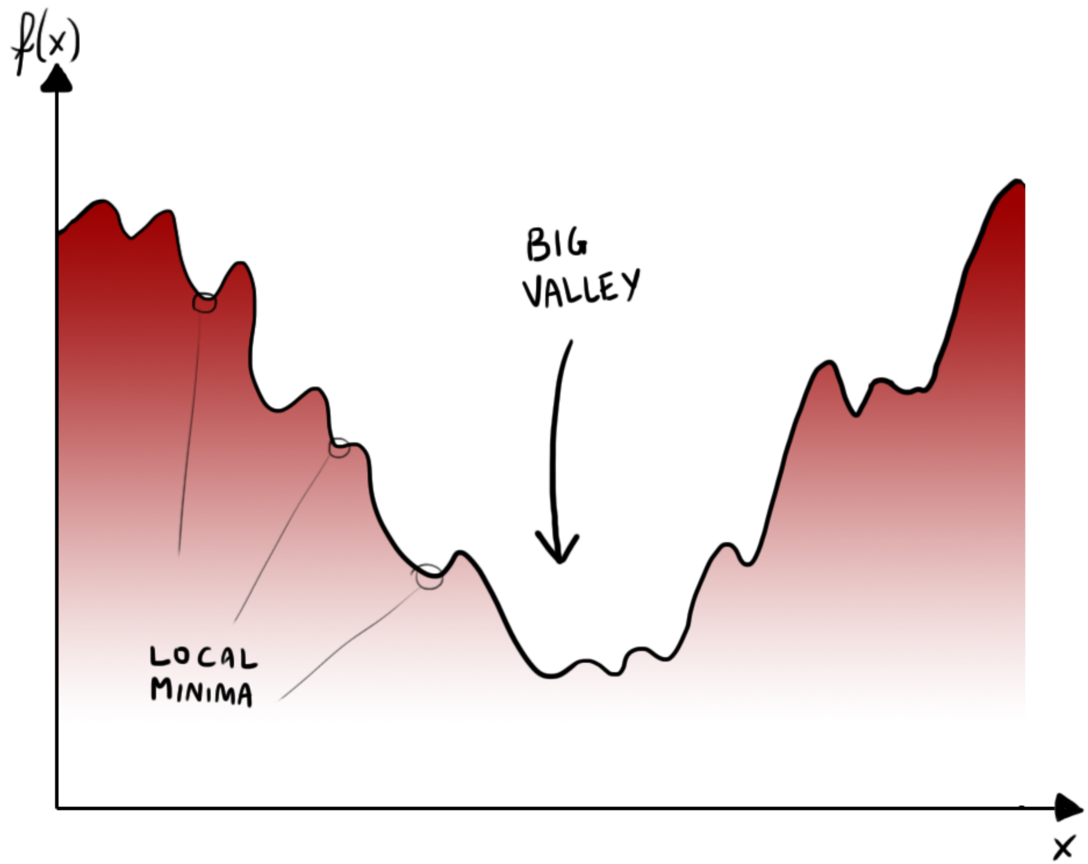
Local optima are not always global optima

- For many optimization problems, a closer approximation to the global optimum is required

- More complex search schemes have to be adopted to balance in an optimal way exploration and exploitation

Attraction basins

- Local minima tend to be clustered (good local minima tend to be closer to other good minima)
- The attraction basin associated with a local optimum is the set of points X which are mapped to the given local optimum by the local search trajectory
- if local search stops at a local minimum, kicking the system to a close attraction basin can be much more effective than restarting from a random configuration



Structure in optimization problems: the “big valley” hypothesis.

Modifications of local search based on perturbations

- local search by small perturbations is an effective technique but additional ingredients are in certain cases needed to obtain superior results

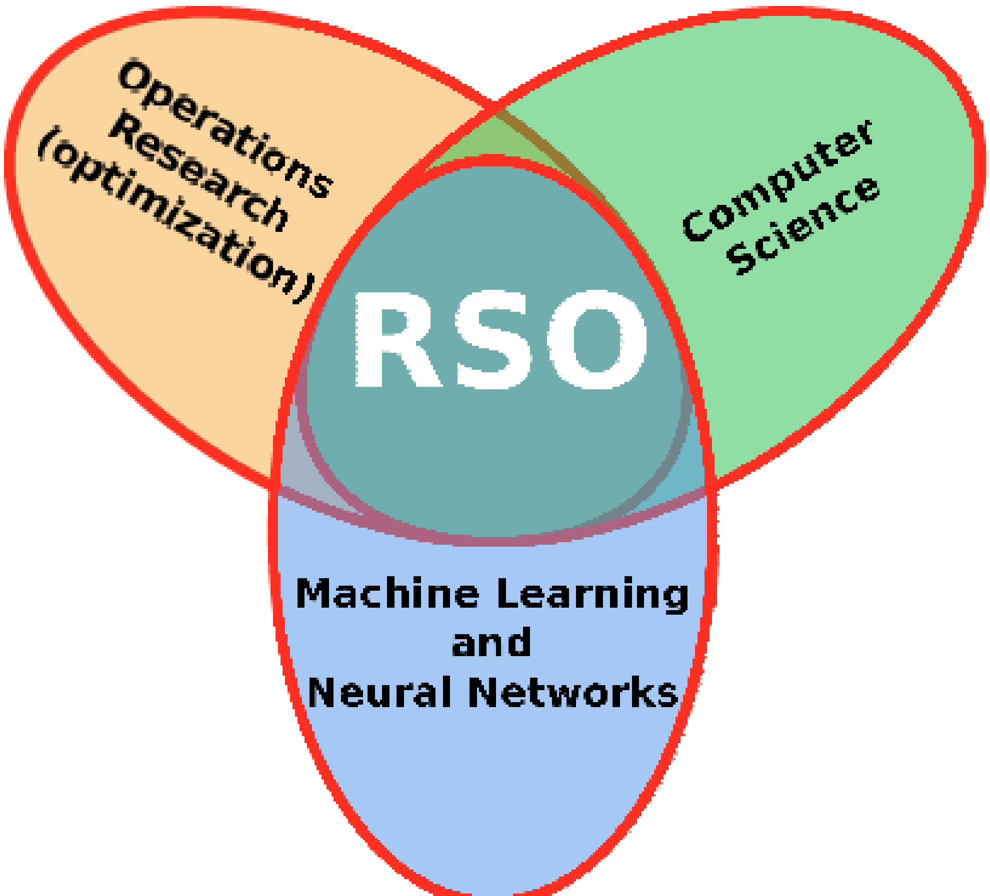
Local search in action: how to build a better bike, from the initial model (left) to a worse variation (middle), to the final and better configuration (right).

Reactive Search Optimization (RSO): Learning while searching

- Many problem-solving methods are characterized by a certain number of choices and free parameters, usually manually tuned.
- Parameter tuning can be automated as a part of the optimization algorithm
- This leads to self-contained, fully automated algorithms, independent from human intervention

Reactive Search Optimization (RSO) integrates online machine learning techniques and search heuristics for solving complex optimization problems.

Reactive Search Optimization (RSO):



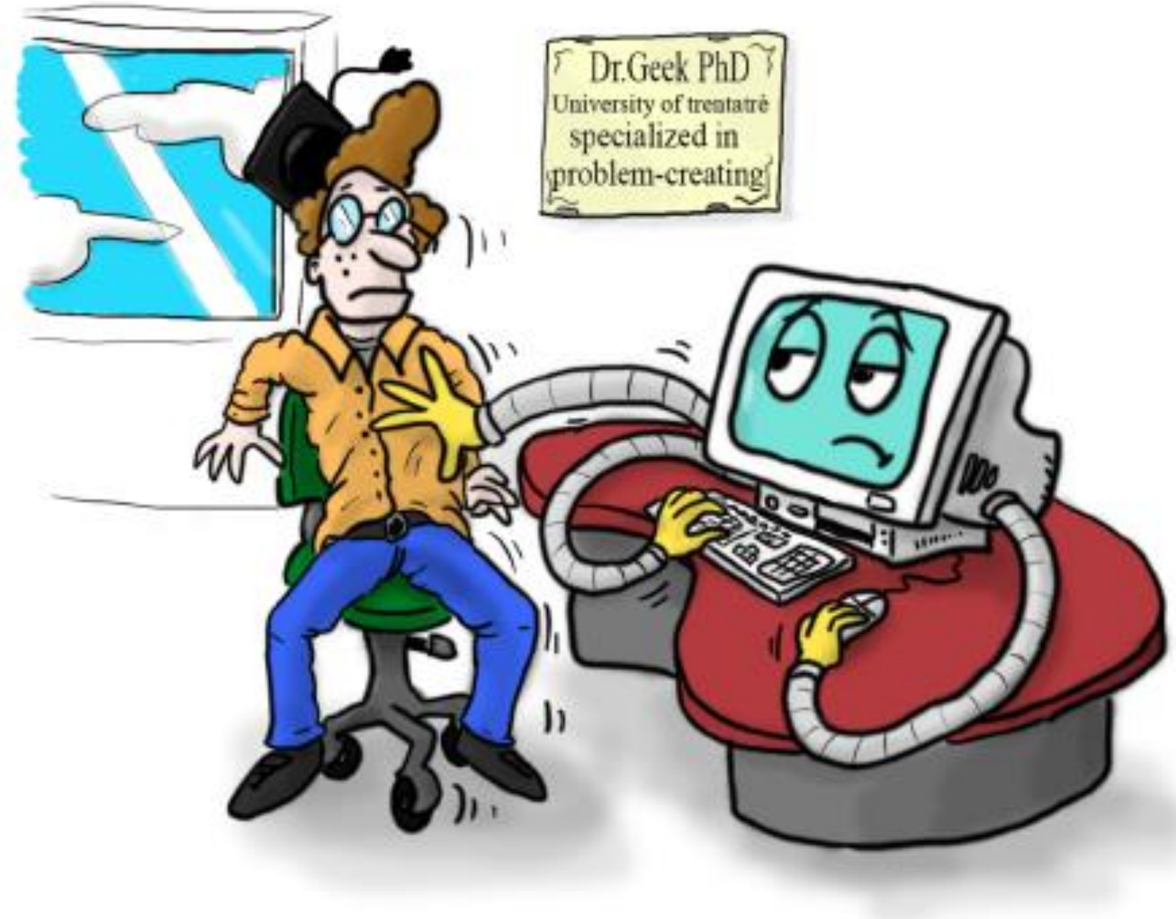
RSO is a the intersection of optimization, computer science (algorithms and data structures) and machine learning.

Reactive Search Optimization

- RSO can be applied to systems that require to set some operating parameters to improve its functionality.
- A simple loop is performed: set the parameters, observe the outcome, then change the parameters in a strategic and intelligent manner until a suitable solution is identified
- In order to operate efficiently, RSO uses memory and intelligence to improve solutions in a directed and focused manner

Reactive Search Optimization

- While many alternative solutions are tested in the exploration of a search space, patterns and regularities appear
- The human brain quickly learns and drives future decisions based on previous observations.
- This is the main inspiration source for inserting online machine learning techniques into the optimization engine of RSO



Algorithms with **self-tuning** capabilities like RSO make life simpler for the final user. Complex problem solving does not require technical expertise but is available to a much wider community of final users

RSO based on prohibitions: tabu search

- Basic idea: using prohibitions to encourage diversification

How?

- While constructing a trajectory for local minima search, every time a move is applied, the inverse move is temporarily prohibited

Tabu search: an example

- Let $\chi=\{0,1\}^L$
- The neighborhood is obtained by applying the elementary moves μ_i, $(i = 1,\dots,L)$ that change the i -th bit of the string $X = [x_1,\dots, x_i,\dots, x_L]$
- At each step, the selected move is the one that minimizes the target f in the neighborhood even if f increases, to exit from local minima.
- As soon as a move is applied, **the inverse move is temporarily prohibited**

Tabu search

- Tabu search can generate cycles. For example, if the current point $X^{(t)}$ is a strict local minimum
- In general, the inverses of the moves executed in the most recent part of the search are prohibited for a period T, in order to avoid cycles and to diversify

Prohibition and diversification

- Let H(X, Y) be the Hamming distance between two strings X and Y
- if only allowed moves are executed, and T satisfies T < (n - 2) (at least two moves are allowed at each iteration), then

- The Hamming distance H between a starting point and successive points along the trajectory is strictly increasing for $T+1$ steps:

$$H(X^{(t+\tau)}, X^{(t)}) = \tau \quad \text{for} \quad \tau \leq T+1.$$

- The minimum repetition interval R along the trajectory is $2(T+1)$:

$$X^{(t+R)} = X^{(t)} \;\Rightarrow\; R \geq 2(T+1).$$

Prohibition and diversification(2)

prohibition is related to the amount of diversification :

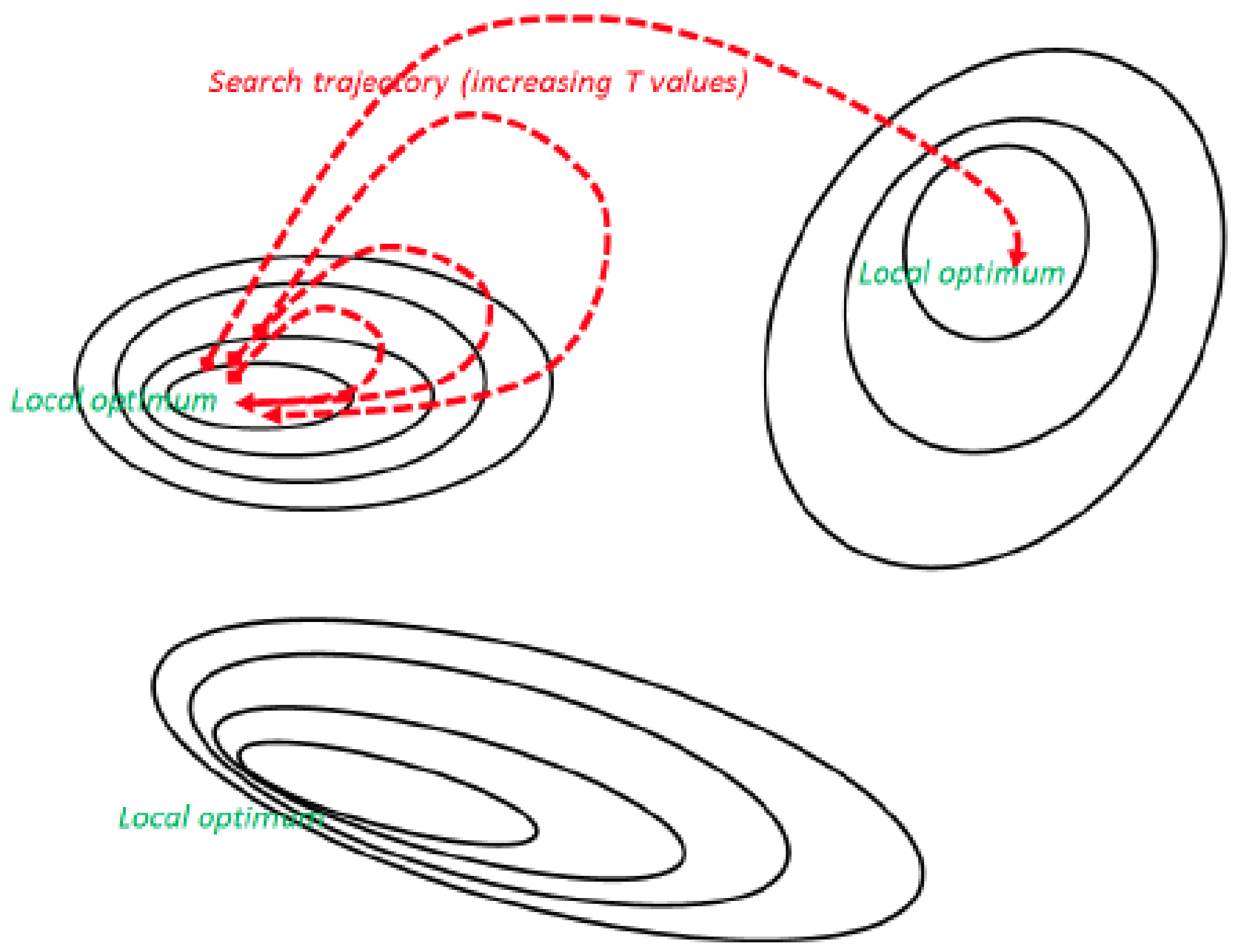
- the larger T , the larger is the distance H that the search trajectory must travel before it is allowed to come back
- If T is too large, the number of allowed moves will shrink, leading to less freedom of movement.

	Iteration t	$X^{(t)}$								$f(X^{(t)})$	$H(X^{(t)},X^{(t)})$
	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	1	1	1
	2	0	0	0	0	0	0	1	1	3	2
	3	0	0	0	0	0	1	1	1	7	3
$T+1$ →	4	0	0	0	0	1	1	1	1	15	4
	5	0	0	0	0	1	1	1	0	14	3
	6	0	0	0	0	1	1	0	0	12	2
	7	0	0	0	0	1	0	0	0	8	1
$2(T+1)$ →	8	0	0	0	0	0	0	0	0	0	0

An example of the relationship between prohibition T , and diversification measured by the Hamming distance H(X(t);X(0)) . T = 3 in the example

Tuning the T parameter

- The parameter T should be tailored to the specific problem
- BUT the choice of a **fixed T** without a priori knowledge is difficult
- RSO uses a simple mechanism to change T during the search so that the value $T^{(t)}$ is appropriate to the local structure of the problem
- RSO determines the minimal prohibition value which is sufficient to escape from an attraction basin around a minimizer



RSO with prohibitions in action. Three locally optimal points are shown together with contour lines of the function to be optimized. When starting from a locally optimal point, RSO executes loops which reach bigger and bigger distances from the attractor, until another attraction basin is encountered (if present).

RSO for tabu search

- T is equal to one at the beginning
- T **increases** if the trajectory is trapped in an attraction basin
- T **decreases** if unexplored search regions are visited, leading to different local optima

RSO: conclusions

- If the problem has a single local optimum the power of RSO is not needed, although not dangerous
- Most real-world problems are infested with many locally optimal points
- RSO is crucial to **transform a local search building block into an effective and efficient solver**.
- RSO with prohibitions has been used for problems ranging from combinatorial optimization to the minimization of continuous functions and to sub-symbolic machine learning tasks

GIST

- Local search is a simple and very effective way to identify improving solutions for discrete optimization problems
- It generates **a sequence of changes, each change being local**
- Local search stops at **locally-optimal points** and the current search trajectory is trapped
- Additional **diversification** means are needed to escape from local attractors.

GIST (2)

- Reactive Search Optimization (RSO) uses **learning and adaptation during the optimization process**, to fine-tune the search technique to the current problem, task and local properties.
- An intelligent module overseeing the basic local search process
- It automatically balances **diversification** and **intensification**