

# Simulated Annealing Tools Software

version 1.03

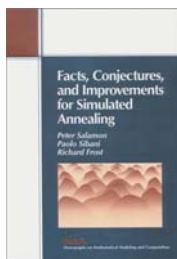
## USER GUIDE

### Table of Contents

1. What is it? .....	1
2. How do I set it up? .....	2
2.1. Completing the installation for Matlab .....	2
3. How do I run an example? .....	3
3.1. For Matlab .....	3
4. How do I code my own problem? .....	4
4.1. For Matlab .....	4
5. Copyright, Trademarks, and License Fees .....	5

### 1. What is it?

A suite of software modules to supplement the book:



P. Salamon, P. Sibani, and R. Frost. [\*Facts, Conjectures, and Improvements for Simulated Annealing\*](#). SIAM Monographs on Mathematical Modeling and Computation. ISBN 0-89871-508-3.

Updates to the software are maintained at <http://www.frostconcepts.com/software/>.

This version is a source code release in

Matlab® (requires version 6)

It consists of 1driver plus many annealing-specific methods and support routines. Like-methods use a common calling interface so that users can easily substitute their own methods for those provided with the package. Examples of applications are also included in the release.

## 2. How do I set it up?

Uncompress the package into a folder or directory on your system.

- On PC's running some form of the Windows® operating system: put the uncompressed "satools" folder in the "Program Files" directory on your "C:" drive.
- On Macintosh computers, put the uncompressed "satools" folder in your "Applications" folder.
- On Unix systems, put the uncompressed directory "satools" in your home directory.

### 2.1. Completing the installation for Matlab

You need to have Matlab version 6 (SR12) or better installed on your computer.

Add the "m" directory of "satools" to your Matlab path. Typically this is done by adding an **addpath** command to your Matlab **startup** file. If you are not sure whether or not you have a `startup.m` file:

1. Start Matlab.
2. By default, the initial directory is probably your working directory. On a PC, this is probably something like `C:\matlab_sv12\work`. To check, try  

```
>> pwd
```

at the Matlab prompt.
3. Edit (or create a new) `startup.m` file in this directory. Add an `addpath` statement for the "m" directory of "satools". On a PC, you might add 3 lines like

```
%  
SATOOLS = 'C:\Program Files\satools\m' ;  
addpath(SATOOLS) ;
```

Save and close the file when you are finished.

4. Run the startup script by typing its name at the Matlab command prompt:  

```
>> startup
```
5. Check that the path was properly added by running the SA Tools version utility function  

```
>> satoolsversion
```

The current version # of `satools` should echo on the screen.

If necessary, see the online Matlab documentation at  
<http://www.mathworks.com/access/helpdesk/help/techdoc/matlab.shtml>  
for more details.

### 3. How do I run an example?

#### 3.1. For Matlab

1. Copy one of the example directories from the `satools/m/examples/` directory to your work area. For example, try the `proteinfold` directory.
2. Start Matlab (if it is not already running) and `cd` into the directory you *copied*. For example  

```
>> cd proteinfold
```
3. Execute the `try_me` function in that directory. For example  

```
>> try_me
```

You should get plenty of output on the screen, starting with something like

```
newstate = sequence_new
X is cell of size 1 4
cost = sequence_cost
...
```

and continuing with a temperature step accounting of the simulation

t	T	<E>	Etarget	Estd	e	steps	Ebsf
0	Inf	0	0	0	0	0	0
1	1e+004	0	0	0	Inf	1	0
2	99	0.25	0.113	0.683	15.8	1	0
...							

then finishing with a listing of an estimate of `rho` and the energy bin parameter values.

The `try_me` function performs two tasks:

1. Assigns values to the input variables of the SA Tools method `anneal`.
2. Calls the `anneal` method.

Since the `anneal` method has a large number of inputs, it is simpler to edit a “driver” function that assigns values to the inputs than to try and maintain them on the command line of an interactive session. Documentation on the input values is currently limited to comments built into the method. To view them, enter the following at the Matlab prompt

```
>> more on
>> help anneal
```

The motivations and use of methods implemented in SA Tools are discussed in the book referenced in section 1 of this document.

## 4. How do I code my own problem?

### 4.1. For Matlab

Version 1.03 and later contain the convenience function `mktemplate` for auto-generating the basic files a user needs to get started.

1. Start Matlab if it is not already running.
2. Go to your work directory.
3. Choose a name for your problem. It should be of moderate length and also suitable for a directory name.
4. Execute

```
>> mktemplate('problem') ;
```

where 'problem' is the selected problem name (in single quotes).
5. A directory with the same name as your problem containing 5 Matlab m-files will be created. Matlab will change its current directory to this new directory.
6. The generated files are fully functional – although the returned cost is simply a random number. To execute, run the familiar function

```
>> try_me
```

The generated files are only a starting point. From here, the user needs to code initialization, instantiation, `moveclass` (perturb), and cost functions. Further, tuning of parameter values for temperature initialization, equilibration, etc. will likely be needed. Documentation on these subjects is currently limited to the book (see section 1 of this document), the code (e.g., try `help hoffmann`), and by way of example: the six example toy applications included with the package.

Happy annealing!

## 5. Copyright, Trademarks, and License Fees

The software discussed here is subject to the following copyright, trademark, and license restrictions.

Copyright (c) 2002, by Richard Frost and Frost Concepts. All rights reserved except where otherwise noted. This software is offered on an "AS IS" basis. The copyright holder(s) provide no warranty, expressed or implied, that the software will function properly or that it will be free of errors. This software may be freely copied and distributed for research and educational purposes only, provided that the above copyright notice appear in all copies. A license is needed for commercial sale or use, in whole or in part, from Frost Concepts. Users of the software agree to acknowledge the copyright holder(s).

Software Licensing:

- Individual or group academic research (non-commercial gain) or Purchase Evaluation: Freely distributed.
- Individual or departmental non-academic use without redistribution or "service bureau" application: \$400 for current version. Updates to current version at no additional cost.
- Unlimited commercial or "service bureau" use by individual or site: \$2000 for current version. Updates to current version at no additional cost.

Fees are payable to:

Frost Concepts  
P.O. Box 721508  
San Diego, CA 92172

"Simulated Annealing Tools Software", "SA Tools", "satoools", and Simulated Annealing eXplorer" are trademarks of Frost Concepts.

*last update: 11/03/2002*