XXI ENMC
ENCONTRO NACIONAL DE MODELAGEM COMPUTACIONAL
IX ECTM
ENCONTRO DE CIÊNCIA E TECNOLOGIA DE MATERIAIS

08 a 11 de Outubro de 2018
Instituto Federal Fluminense
Búzios - RJ

A SCILAB FRAMEWORK TO TEST METAHEURISTICS: LAST RESULTS

Gustavo Mendes Platt[1] - e-mail: gmplatt@furg.br
Douglas Alves Goulart[2] - e-mail: douglasagoulart@gmail.com
[1]Universidade Federal do Rio Grande – Santo Antônio da Patrulha, RS, Brazil
[2]Universidade Federal do Rio Grande – Rio Grande, RS, Brazil

Abstract. *Many real-world optimization problems have been solved by metaheuristics in the last decades, mainly as a consequence of the increasing capacity of personal computers. In the same context, the number of available metaheuristics is huge. Furthermore, countless engineering problems are described as minimization/maximization procedures with adequate objective functions. In this scenario, the comparison between metaheuristics is frequently addressed in the specialized literature. In this work, we presented a modular framework (developed in Scilab language) devoted to the test of metaheuristics in several engineering problems (mainly in Chemical and Mechanical Engineering problems; as example we can cite high pressure phase equilibrium problems and inverse robot kinematics problem). The framework permits the selection of the metaheuristic itself as well as the control parameters (stopping criterion, number of individuals, specific parameters for each algorithm). Moreover, the computation structure presents a statistical comparison between the selected algorithms with respect to the number of function evaluations, iterations and computational time (including mean, standard deviation and a non-parametric statistical test), with minimal user-interference. New metaheuristics can be added to the framework using a standard coding procedure. The results are displayed in graphical form and can be exported using csv standard. Finally, we consider that this kind of computational structure is useful in the standardization of the procedures employed in the comparisons of these stochastic algorithms.*

Keywords: *metaheuristics, engineering optimization, phase equilibrium*

INTRODUCTION

Metaheuristics have been applied in countless engineering problems in the last decades. In this scenario, we observed a proposition of a massive quantity of new metaheuristics – mimetizing swarm intelligence of species, natural phenomena (lightning, wave movement, for instance), social behavior, among others. These algorithms are usually tested in some benchmarks functions (Himmelblau function, Ackley function etc.), but the use of real-world problems as "hard" tests for this class of algorithms has been advocated in recent works.

In this work we detailed the construction of a framework devoted to the evaluation of metaheuristics in Engineering problems (mainly in Chemical and Mechanical Engineering fields). This computational structure permits the comparison between algorithms, as well as the evaluation of a single method in different problems. Furthermore, the framework is also useful in parameter tuning of the algorithms, permitting a rapid screening of relevant control parameters.

Methodology

We used an arbitrary choice for the pair (metaheuristic, problem) in order to demonstrate the capabilities of the framework. Thus, we have chosen the Symbiotic Organisms Search (SOS) (Cheng & Prayogo, 2014) as the tested algorithm and two thermodynamic calculations – the estimation of binary interaction parameters for the Wilson model and the calculation of a reactive azeotrope – as the engineering problems.

Description of the Metaheuristic

The Symbiotic Organisms Search is a recently proposed algorithm (Cheng & Prayogo, 2014), which was tested in several engineering fields (Platt, 2016; Tejani et al., 2016; Panda & Pani, 2016). The SOS algorithm is based on three "operators" (emulating the interaction of individuals in an environment): mutualism, commensalism and parasitism. The algorithmic structure of SOS (not shown in detail here) can be found elsewhere Cheng & Prayogo (2014); Platt (2016).

The SOS shows a desirable property: a low number of control parameters. Here, we are interested mainly in the control parameters of the method: eco_size and max_iter (a third control parameter, max_fit_eval – representing the maximum number of function evaluations – was not considered in this example; in fact, we assign a large value to max_fit_eval and the algorithm stops always by max_iter). The parameter eco_size represents the number of individuals in the population, and max_iter is the maximum number of iterations of a single run (used as a stopping criterion).

Description of the Engineering Problems

Parameter Estimation Parameter estimation of thermodynamic models is an extremely important task in separation engineering, since the correct behavior of the thermodynamic model is severely dependent on the quality of the parameters. Here, we used, as an example, the parameter estimation of the Wilson model for the binary pair *tert*-butanol (1) + 1-butanol (2). A detailed description of the problem can be found, for instance, in Platt (2016).

The objective-function – to be minimized – is represented by:

$$f\left(\theta_1,\theta_2\right)=\sum_{j=1}^{np}\sum_{i=1}^{2}\left(\frac{\gamma_{i,j}^{exp}-\gamma_{i,j}^{calc}\left(\theta_1,\theta_2\right)}{\gamma_{i,j}^{exp}}\right)^2, \tag{1}$$

where γ_i refers to the activity coefficient of component i calculated by the Wilson model, the superscripts exp and $calc$ represent, respectively, experimental and calculated data and np is the number of experimental points, published by Wisniak & Tamir (1976).

This problem exhibits one local minimum and one global minimum; this characteristic was studied by several authors (see, for instance, Alvarez et al. (2008); Gau et al. (2000)), with

Table 1- Global and local minima in parameter estimation problem (Platt, 2016).

	θ_1	θ_2	f
Global minimum	-865.1	2419.9	0.011
Local minimum	852.2	-607.8	0.033

Table 2- Global minima in the reactive azeotropy problem (Platt, 2016).

Component	x_i^{az1}	y_i^{az1}	T^{az1}	x_i^{az2}	y_i^{az2}	T^{az2}
isobutene (1)	0.0446038	0.1727712	392.2444	0.0138208	0.0748231	391.17559
methanol (2)	0.1197971	0.2378773	-	0.4037263	0.4406101	-
MTBE (3)	0.8355991	0.5893516	-	0.5824529	0.4845668	-

different approaches. Thus, we are interested in the capability of the algorithm to obtain, as much as possible, the global minimum. The minimum points of the problem are presented in Table 1.

This problem is essentially unconstrained, since the values of the binary parameters (referred here as θ_1 and θ_2) do not obey any physical constraints. The initial values for the elements in the population are randomly generated in the interval $[-5000, 5000] \times [-5000, 5000]$.

Finally, we are dealing with a two-dimensional problem and we can observe the performance of the algorithm using a graphical tool.

Reactive Azeotropy The reactive azeotropy problem focused here was analyzed by several authors (see, for instance, Platt (2016)). The problem is described by a nonlinear set of algebraic equations, converted into an optimization problem through a scalar function representing the sum of the squares of the residues (for each nonlinear equation). This problem involves three components: isobutene, methanol and methyl-*tert*-butyl-ether (MTBE). The variables (to be found) are the molar fractions of the liquid phase (x_i), the molar fraction of the vapor phase (y_i) and the system temperature (T). The system pressure is specified at 8 atm. The two solutions of the problem (the two azeotropes) are represented by az_1 and az_2. Table 2 contains the coordinates of the two reactive azeotropes (obviously, the two solution are global and null minima of the problem).

The Construction of the Framework

The computational framework was developed in Scilab[1] language, an open-source platform with built-in functions for linear algebra and matrix manipulation.

[1] ©2011-2015, Scilab Enterprises

In order to facilitate the data entry by the user, we constructed a GUI (*Graphical User Interface*). The GUI was built using the GUI builder toolbox (developed by Luh & Violeau (2014)). All metaheuristic algorithms are programmed as functions in Scilab environment, permitting the addition of new methods with minimal changes in the structure of the framework. The same methodology was employed in the construction of the real-world problems, i.e., new problems can be added to the code maintaining the same algorithmic structure.

In the current version, the computational code permits the selection of the algorithm, the problem to be solved and several control parameters for each metaheuristic. Furthermore, the user can select the number of runs (in order to produce statistical results) and – for two-dimensional problems – an option for a "graphical run". In this last situation, the evolution of the algorithm is displayed in a graphical form. Without the "graphical run" option, only the best element of the population for each run is presented (as a red circle) at the end of the iterations.

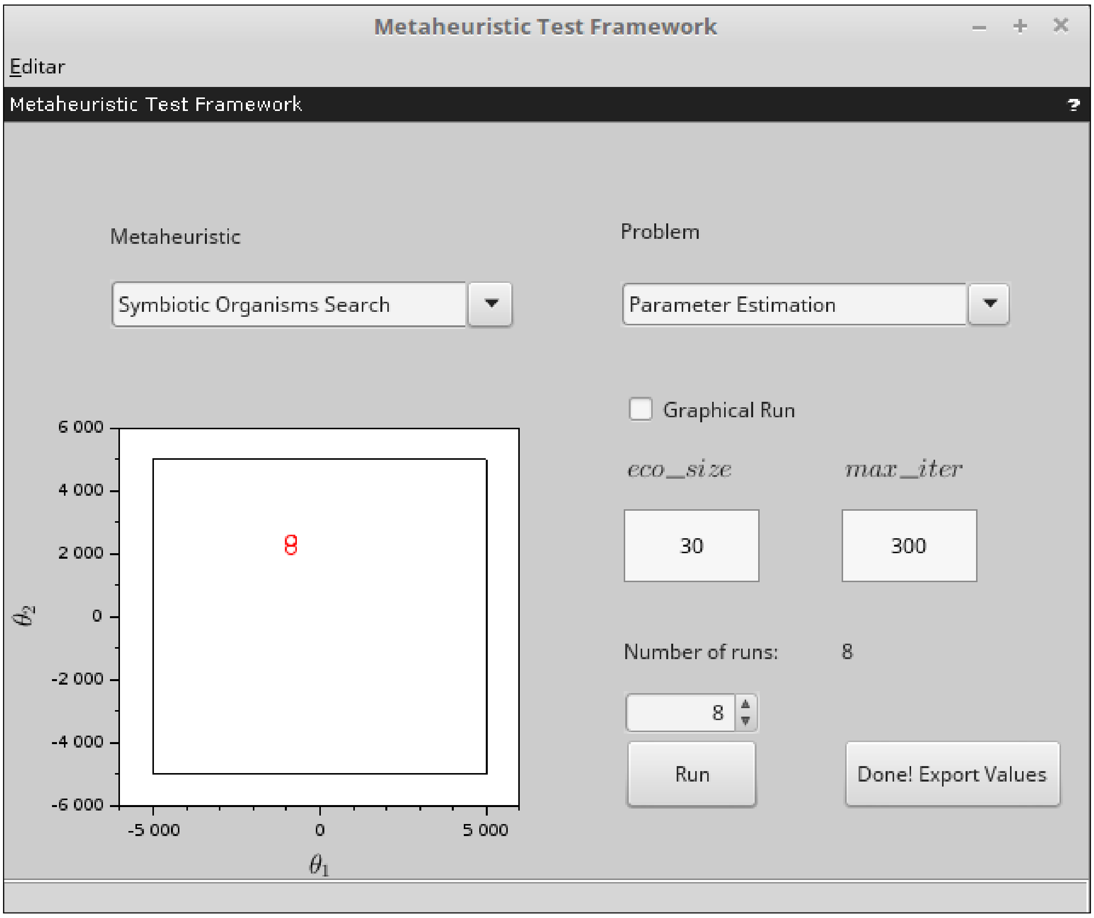


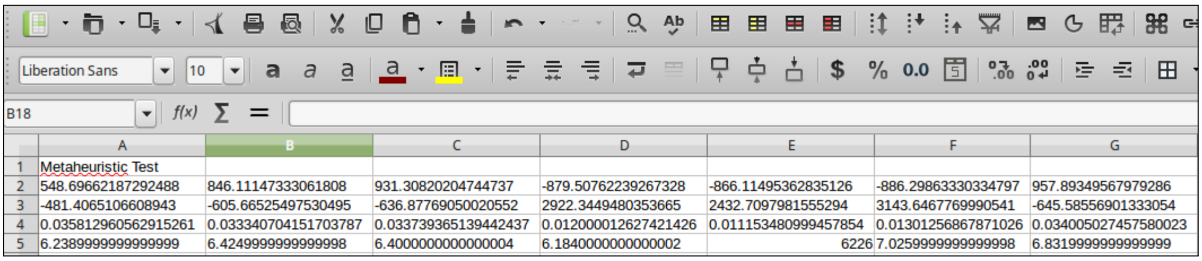
Figure 1- GUI of Metaheuristic Test Framework.

Numerical Results

First Test – Parameter Setting of SOS

The capabilities of the developed framework will be illustrated in the adjustment of control parameters of SOS – eco_size and max_iter – in the determination of global optimum of the thermodynamic problem. As pointed by Platt & Lima (2018), the parameter setting is a fundamental step in the use of metaheuristics, since the quality of the control parameters can drastically affect the performance of the algorithm.

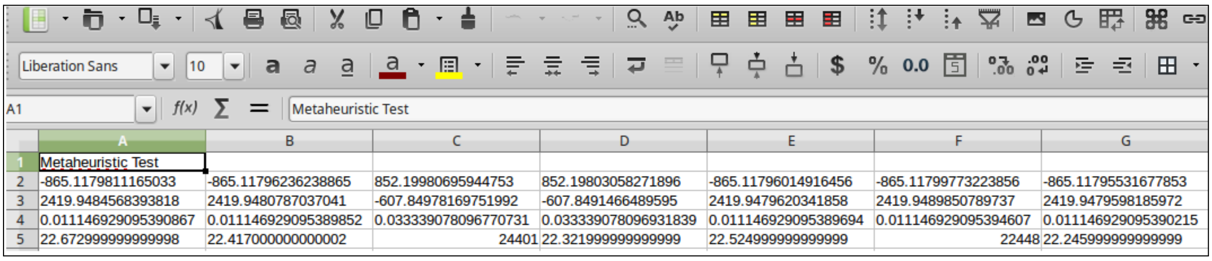
Figure 2 presents the results of SOS (exported for LibreOffice Calc[2], using a comma-separated-value – csv – standard) considering $eco\_size = 10$ and $max\_iter = 100$. We noted a low accuracy of values, in a comparison with Table 1. Moreover, the local minimum was identified in the most part of runs. Considering this situation, the parameter setting procedure indicates an increase in the number of max_iter (in order to enhance the accuracy of the final values).



	A	B	C	D	E	F	G
1	Metaheuristic Test						
2	548.69662187292488	846.11147333061808	931.30820204744737	-879.50762239267328	-866.11495362835126	-886.29863330334797	957.89349567979286
3	-481.4065106608943	-605.66525497530495	-636.87769050020552	2922.3449480353665	2432.7097981555294	3143.6467769990541	-645.58556901333054
4	0.035812960562915261	0.033340704151703787	0.033739365139442437	0.012000012627421426	0.011153480999457854	0.01301256867871026	0.034005027457580023
5	6.2389999999999999	6.4249999999999998	6.4000000000000004	6.1840000000000002	6226	7.0259999999999998	6.8319999999999999

Figure 2- Exported values (θ_1, θ_2, objective-function and computation times) for $eco\_size = 10$ and $max\_iter = 100$.

Figure 3 contains the same objects of Figure 2, but now considering $max\_iter = 500$. We noted a reduction in the values of the objective-function. Besides, the global minimum was found in 5 (of 7) runs.



	A	B	C	D	E	F	G
1	Metaheuristic Test						
2	-865.1179811165033	-865.11796236238865	852.19980695944753	852.19803058271896	-865.11796014916456	-865.11799773223856	-865.11795531677853
3	2419.9484568393818	2419.9480787037041	-607.84978169751992	-607.8491466489595	2419.9479620341858	2419.9489850789737	2419.9479598185972
4	0.011146929095390867	0.011146929095389852	0.033339078096770731	0.033339078096931839	0.011146929095389694	0.011146929095394607	0.011146929095390215
5	22.672999999999998	22.417000000000002	24401	22.321999999999999	22.524999999999999	22448	22.245999999999999

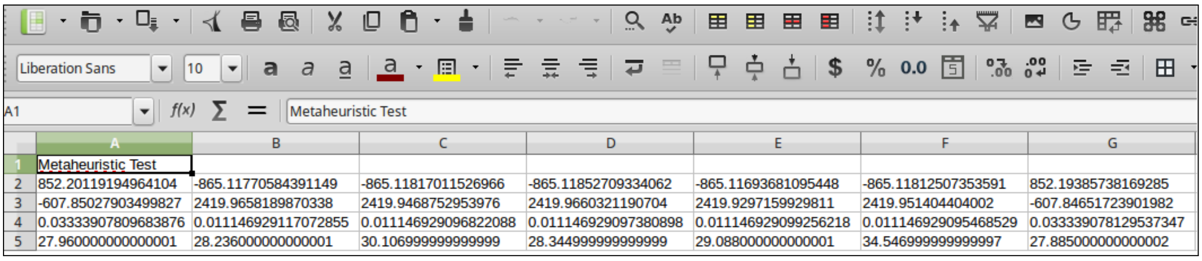
Figure 3- Exported values (θ_1, θ_2, objective-function and computation times) for $eco\_size = 10$ and $max\_iter = 500$.

Now, we will illustrate some effect of $eco\_size = 10$. Increasing this value to 15, we obtain the results of Figure 4. In this situation, we do not observe better results when comparing to $eco\_size = 10$. Obviously, in real parameter setting procedures, the number of runs must be greater than 7. The use of 7 runs is merely illustrative.

Finally, the last test is conducted with $eco\_size = 100$ and $max\_iter = 500$, as detailed in Figure 5. This extreme situation (with a large population in a low-dimension problem) is useful to demonstrate an undesired effect in large populations: some kind of "dispersion" of the individuals in the vicinities of the global optimum. Clearly, the accuracy of the global optimum is lower in this case, even considering that only the neighborhood of the global optimum was identified. The computational cost was also huge when comparing to the previous tests.

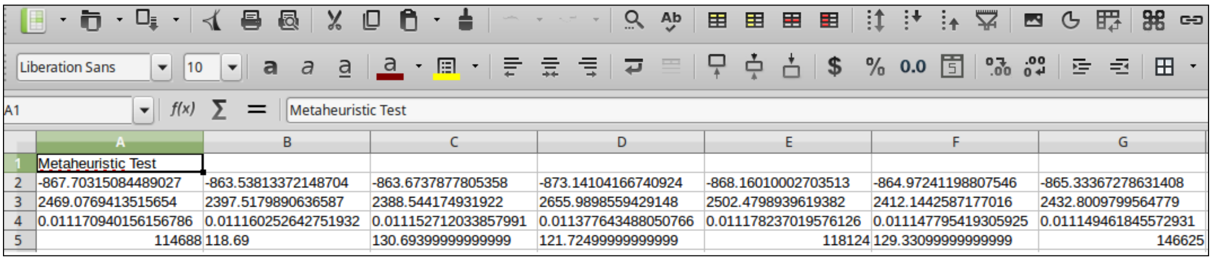
Figure 6 presents, only as a illustration, a typical run of the SOS algorithm in the parameter estimation problem.

[2]©2000-2016, The Document Foundation, Debian and Ubuntu



	A	B	C	D	E	F	G
1	Metaheuristic Test						
2	852.20119194964104	-865.11770584391149	-865.11817011526966	-865.11852709334062	-865.11693681095448	-865.11812507353591	852.19385738169285
3	-607.85027903499827	2419.9658189870338	2419.9468752953976	2419.9660321190704	2419.9297159929811	2419.951404404002	-607.84651723901982
4	0.03333907809683876	0.011146929117072855	0.011146929096822088	0.011146929097380898	0.011146929099256218	0.011146929095468529	0.033339078129537347
5	27.960000000000001	28.236000000000001	30.106999999999999	28.344999999999999	29.088000000000001	34.546999999999997	27.885000000000002

Figure 4- Exported values (θ_1, θ_2, objective-function and computation times) for $eco\_size = 15$ and $max\_iter = 500$.



	A	B	C	D	E	F	G
1	Metaheuristic Test						
2	-867.70315084489027	-863.53813372148704	-863.6737877805358	-873.14104166740924	-868.16010002703513	-864.97241198807546	-865.33367278631408
3	2469.0769413515654	2397.5179890636587	2388.544174931922	2655.9898559429148	2502.4798939619382	2412.1442587177016	2432.8009799564779
4	0.011170940156156786	0.011160252642751932	0.011152712033857991	0.011377643488050766	0.011178237019576126	0.011147795419305925	0.011149461845572931
5	114688	118.69	130.69399999999999	121.72499999999999	118124	129.33099999999999	146625

Figure 5- Exported values (θ_1, θ_2, objective-function and computation times) for $eco\_size = 100$ and $max\_iter = 500$.

Second Test – Application of SOS in the reactive azeotropy problem

The second test is the use of the framework in the reactive azeotropy problem described in a previous section. This problem exhibits two solutions, and the metaheuristic was capable to find only one solution in each run (considering the use of the algorithm without any strategy to find more than one solution). Moreover, since we are dealing with five variables, the framework exhibits projections in several planes, as detailed in Figure 7.

A typical run is presented in Figure. 8. Clearly, the coordinates found by the metaheuristic are close to that presented by Table 2 for az_2. Furthermore, the fitness value (or the objective-function) is close to zero, as expected. It must be noted that for reactive azeotropy problems the equality of compositions is not verified.

CONCLUSIONS

In this work we described the construction of a computational modular framework devoted to test and comparisons of metaheuristics in real-world problems. The computational tool was developed in Scilab language and was applied – as an example – in the parameter setting of a recently proposed algorithm (Symbiotic Organisms Search) in a thermodynamic problem (parameter estimation of Wilson model) with more than one solution. The developed structure was also used in the calculation of a reactive azeotrope in a ternary mixture.

The use of Scilab platform is justified by the built-in tools for linear algebra, the excellent graphical capability – which permitted the creation of a GUI – and by the fact that Scilab is a open-source tool.

The parametrization of metaheuristic is a exhaustive task for many problems, since it involves some alteration in the computational code. The use of the framework made the process easier, considering that the results can be obtained without the manipulation of the algorithms.

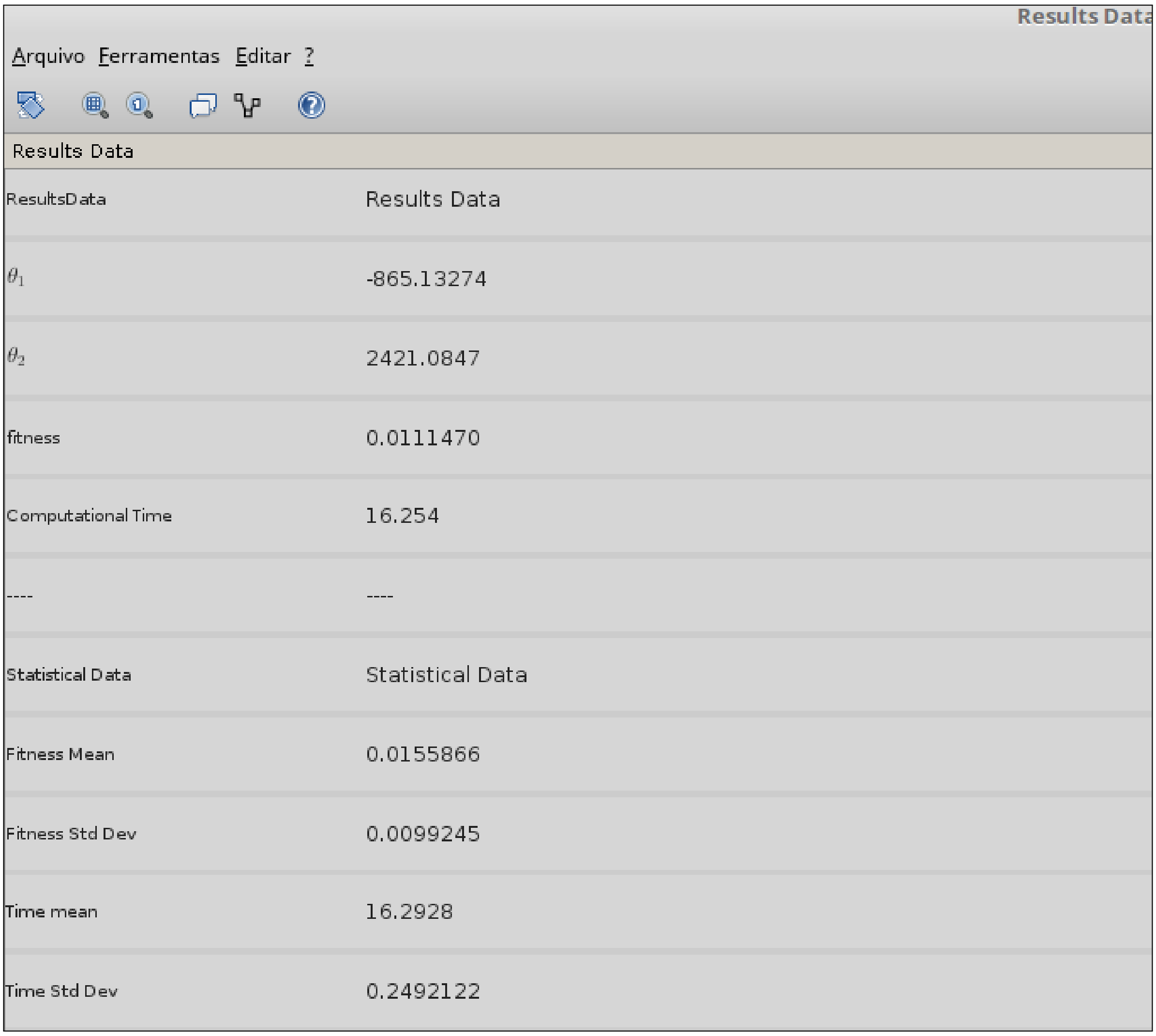


Figure 6- Typical results for parameter estimation problem.

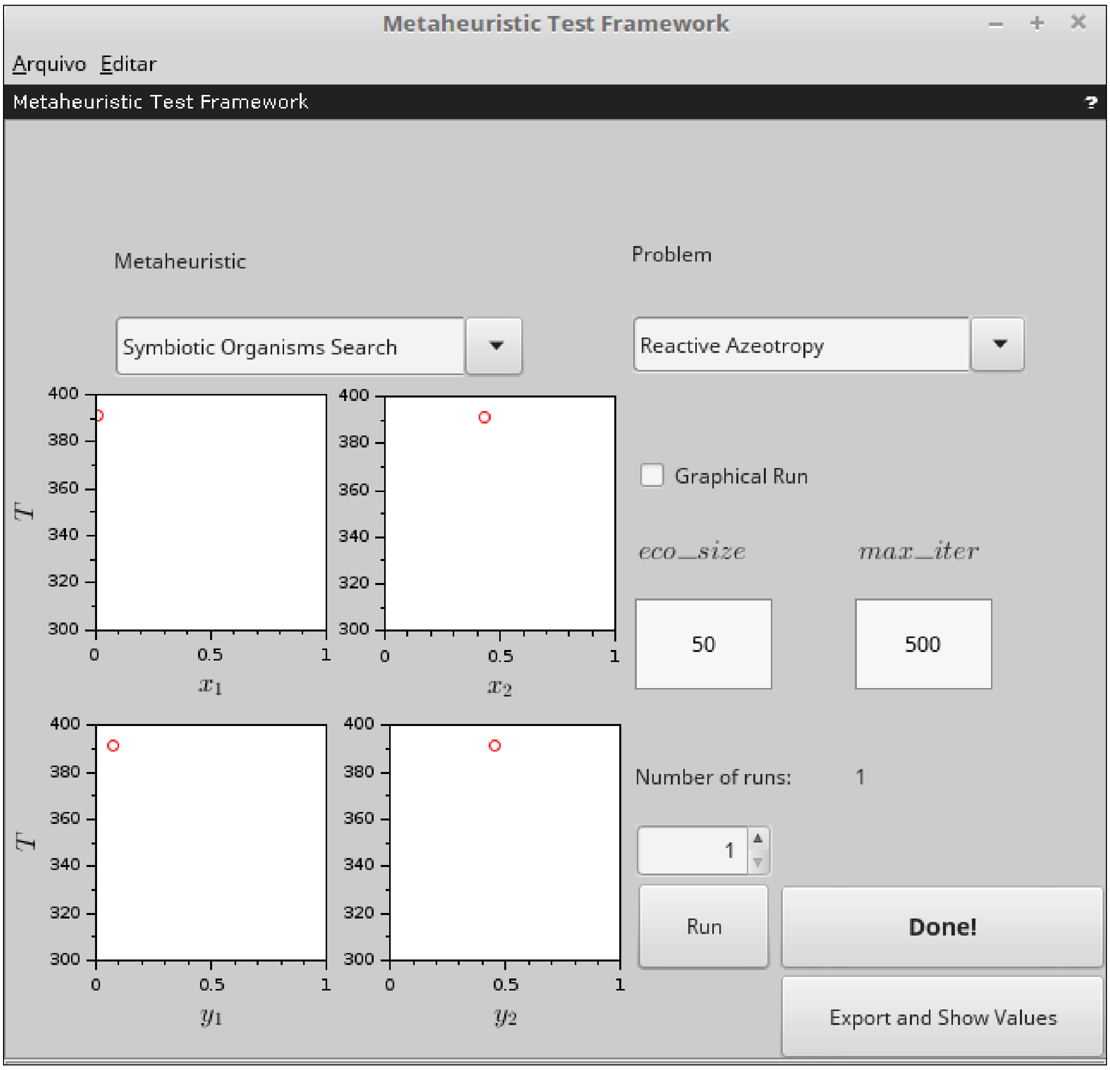


Figure 7- The framework in the reactive azeotropy problem.

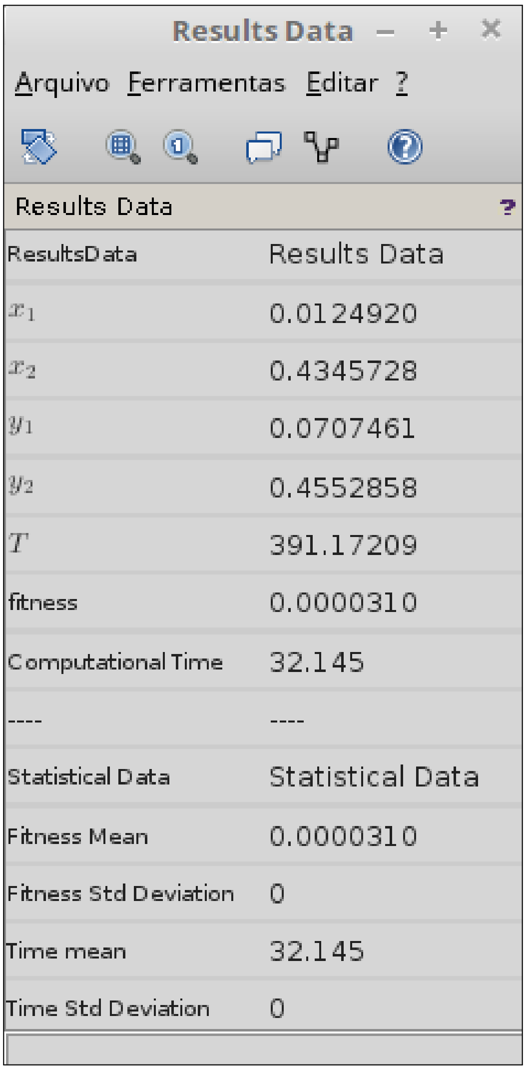


Figure 8- Typical results for reactive azeotropy problem.

Furthermore, the results for different sets of parameters can be exported to spreadsheets.

The results indicated that the framework can be an useful tool for rapid tests in parameter setting, testing and comparison between metaheuristics in different real-world problems.

REFERENCES

Cheng, M.-Y. and Prayogo, D. (2014), Symbiotic Organisms Search: a new metaheuristic optimization algorithm. Computers and Structures, 139, 98-112.

Platt, G.M (2016), Numerical experiments with new metaheuristic algorithms in phase equilibrium problems, International Journal of Mathematical Modelling and Numerical Optimisation, 7(2), 189-211.

Tejani, G.G.; Savsani, V.J. and Patel, V.K. (2016), Adaptive symbiotic organisms search (SOS) algorithm for structural design optimization. Journal of Computational Design and Engineering, 3(3), 226-249.

Panda, A. and Pani, S. (2016), A Symbiotic Organisms Search algorithm with adaptive penalty function to solve multi-objective constrained optimization problems. Applied Soft Computing, 46, 344-360.

Wisniak, J. and Tamir, A. (1976), Correlation of boiling points of mixtures. Chemical Engineering Science, 31, 631-635.

Alvarez, V.H.; Larico, R.; Ianos, Y. and Aznar, M. (2008), Parameter estimation for VLE calculation by global minimization: the Genetic Algorithm. Brazilian Journal of Chemical Engineering, 25(2), 409-418.

Gau, C.-Y.; Brennecke, J. F. and Stadtherr, M. A. (2000), Reliable nonlinear parameter estimation in VLE modeling. Fluid Phase Equilibria, 168, 1-18.

Luh, T.C. and Violeau, D. (2014), GUI Builder version 3.0.

Platt, G.M. and Lima, L.V.P.C. (2018), Azeotropy in a refrigerant system: a useful scenario to test and compare metaheuristics. International Journal of Metaheuristics, 7(1), 43-66.