ATG Research Kit

Globe Optimizer™ Engine

Executive Summary

July 2018

MOTIVATIONS

The *coordinated brute force* method, as introduced in our *Appetizer* "Globe Optimizer," has a unique property: it is originally an exact solution to hard computational problems, but it can be easily scaled back to a heuristic method only by decreasing the amount of the required space (memory). So it is very desirable to see the performance of this method when it is scaled back as a heuristic one. It is what this research kit is about. To this end, we selected *traveling salesman problem* as a strong NP-C problem without any approximation scheme [1] to see the approximation rate of the proposed heuristic method in some practical samples of the TSP. In this regard, we developed the coordinated brute force method to solve any symmetric and asymmetric TSP sample which is compatible with the TSPLIB format. This development has two advantages. Firstly, it can be used as a benchmarking tool to evaluate the capability of the coordinated brute force method as a heuristic method. Secondly, it can be used as the base development for solving many other types of combinatorial optimization problems including *facility location*, *knapsack*, *bin packing*, and many mores. This is why we call it Globe Optimizer™ Engine.

IMPLEMENTATION

We know that as we decrease the size of the method required space, the probability of producing a more inexact solution is increased [2]. Also, we know that the required space of the method is proportional to the width and the length of the space tape [2]. As the length of the tape is equal to the length of the shortest path, we have no control over it. In contrast, we can fix the width of the tape to a desired value without affecting the basis of the method. As this fixation causes the occurrence of *gray eliminations* [2], finding some proper *gray elimination rules* [2] is very important. To this end, we developed two different versions of the proposed method. In 2008, we developed the first version with first generation gray elimination rules. In 2018, the second version with more effective gray elimination rules and parallel processing capabilities was developed. At both versions, the width of the tape was fixed to k, where k is an arbitrary positive integer. In the first version, k was fixed to 1 and 2, but in the second version, it can be selected by user between 1 and 50 for each sample.

The most limited form of the space tape is when $k = 1$, and subsequently when $k = 2$. In terms of accuracy, these forms are the most inaccurate forms of the method, but in terms of speed, they are the fastest forms. In 2008, these two forms were used for benchmarking to see how much reliable is the method in nature, by seeing the results in its weakest form; and how much fast the method could be, by seeing the computation time in its fastest form.

So we selected some sample problems from TSPLIB, where the exact solutions of the samples are known, and therefore, it is possible to calculate the approximation rates.

We selected 24 samples, 14 symmetric and 10 asymmetric. The average size (number of cities) of asymmetric samples in the TSPLIB is smaller than the symmetric ones. It is due to the more complication in the calculation of exact solution in asymmetric TSP.

In 2018, as mentioned before, we set *k* as a parameter in the implementation and it can be selected in the user interface. In addition, we published the binary version of the implementation for free to give the ability of direct testing to the interested users.

RESULTS

In 2008, we run the first version of the method on a single-core Intel Pentium M 750 machine (6050 MIPS), with 533 MHz front side bus and 1 GB of RAM. We set the *k* to 1 and 2, respectively. The following tables show the results including the name of the sample problem in the TSPLIB, the length of the exact solution, the length of the resulted shortest tours for *k* = 1 and 2, the approximation rates, and the CPU consumed time in seconds.

Sample name	**Exact solution**	**Solution for *k* =1**	**Approximation rate for *k* = 1**	**CPU consumed time**	**Solution for *k* =2**	**Approximation rate for *k* = 2**	**CPU consumed time**
a280	2,579	3,425	1.32	52 s	3,403	1.31	110 s
berlin52	7,542	8,344	1.10	2 s	7,955	1.05	4 s
ch150	6,528	8,035	1.23	17 s	7,854	1.20	33 s
d2103	80,450	102,640	1.27	1,410 s	100,008	1.24	3,798 s
eil101	629	816	1.29	1 s	780	1.24	3 s
fnl4461	182,566	248,721	1.36	26,376 s	246,078	1.34	58,125 s
kroA200	29,363	37,278	1.27	121 s	36,302	1.23	345 s
kroB200	29,437	36,732	1.24	123 s	37,368	1.27	336 s
pcb1173	56,892	78,063	1.37	708 s	77,701	1.36	1851 s
rat195	2,323	2,965	1.27	10 s	2,709	1.16	18 s
rat783	8,806	11,870	1.34	2,424 s	11,699	1.32	5,490 s
st70	675	868	1.28	1 s	799	1.18	1 s
ts225	126,643	147,961	1.16	1,095 s	142,659	1.12	2,010 s
tsp225	3,919	4,969	1.26	25 s	4,874	1.24	46 s

Table 1- Results for symmetric samples

Sample name	Exact solution	Solution for k =1	Approximation rate for k = 1	CPU consumed time	Solution for k =2	Approximation rate for k = 2	CPU consumed time
br17	39	47	1.20	1 s	45	1.15	1 s
ft53	6,905	8,344	1.20	2 s	8,040	1.16	4 s
ftv33	1,286	1,497	1.16	1 s	1,406	1.09	1 s
ftv38	1,530	1,757	1.14	1 s	1,633	1.06	1 s
ftv70	1,950	2,298	1.17	2 s	2,330	1.19	2 s
ftv170	2,755	3,589	1.30	8 s	3,382	1.22	15 s
kro124	36,230	41,928	1.15	29 s	40,988	1.13	64 s
rbg323	1,326	1,673	1.26	35 s	1,615	1.21	57 s
rbg403	2,465	3,253	1.32	76 s	3,160	1.28	163 s
rbg443	2,720	3,783	1.39	120 s	3,713	1.36	245 s

Table 2- Results for asymmetric samples

In 2018, we did not publish any test results as the interested users can download the binary version for both of the Microsoft Windows and Linux platforms and apply the method to their own sample problems and see the results in more accurate forms by setting k up to 50. The binary version supports parallel processing and benefits multithreading and can run on as much as CPU cores that the operating system can support. So the CPU consumed time is notably smaller than the numbers you see in the above tables. Another interesting point here is that the coordinated brute force method treats both of the symmetric & asymmetric samples in a same way, i.e., exactly the same algorithm is used for solving both of the symmetric & asymmetric samples. It lets the user apply the method easily to asymmetric samples with the size larger than 1000 nodes, which is a great challenge for other currently known methods.

THE KIT MATERIALS

In addition to this executive summary, this research kit contains a research report which explains the technical details of how to use the coordinated brute force method in its heuristic form and the gray elimination rules which are used in our implementations. This research kit also includes the documented source code of our 2018 implementation. Although this implementation is multithreaded, but we have also added a single-thread version of it to make it easy for the kit users to understand the nature of our implementation. So we have two source codes in this research kit of exactly the same method, one in single-thread mode and the other in multithread mode. All source codes are written in Free Pascal Language, a free cross platform Pascal compiler. They can be directly compiled by Lazarus (a free cross platform IDE for Free Pascal). We have also included the binaries of the multithreaded implementation for the Windows and Linux platforms in this research kit.

LICENSING

This research kit is under ATG Blue License which entails:

Redistribution and use in source and binary forms and output resulted from the binary form, with or without modification, complete or partial, are permitted provided that the following four conditions are met:

1- Redistributions of source code, free or priced, are prohibited except having a written permission from Aria Technology Group.

2- Redistributions in binary form or output resulted from the binary form, if applicable, should not retain the Aria Technology Group copyright notice and the Aria Technology Group URL address, and the distributor, real or legal person, can have his/her own licensing rules and pricing policy.

3- Documents included in a kit, are named as source materials and exactly treated as what mentioned in the first condition (condition 1). Documents may include PDF files, text files, images, audios, and videos.

4- This license is automatically transferred to the subsidiary(s) of the licensee party, but not to the parent organization (if the parent is a third real or legal person) of the licensee party.

THE SOFTWARE (AND THE INTELLECTUAL MATERIAL) IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL ARIA TECHNOLOGY GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE (AND THE INTELLECTUAL MATERIAL) OR THE USE OR OTHER DEALINGS IN THE SOFTWARE (AND THE INTELLECTUAL MATERIAL).

REFERENCES

[1] C. H. Papadimitriou, "*Computational Complexity*," First Ed. Addison Wesley, 1994.
[2] F. Laleh, "Globe Optimizer," *ATG Press Appetizer,* Vol. 1, pp. 18-24, 2005.