Using Local Information To Guide Ant Based Search

Simon Kaegi and Tony White

School of Computer Science, Carleton University 1125 Colonel By Drive Ottawa, Ontario, Canada K1S 5B6 { skaegi, arpwhite } @scs.carleton.ca

Abstract. Marco Dorigo et al. used Ant System (AS) to explore the Symmetric Traveling Salesman Problem and found that the use of a small number of elitist ants can improve algorithm performance. The elitist ants take advantage of global knowledge of the best tour found to date and reinforce this tour with pheromone in order to focus future searches more effectively. This paper discusses an alternative approach where only local information is used to reinforce good tours thereby enhancing the ability of the algorithm for multiprocessor or actual network implementation. In the model proposed, the ants are endowed with a memory of their best tour to date. The ants then reinforce this "local best tour" with pheromone during an iteration to mimic the search focusing of the elitist ants. The environment used to simulate this model is described and compared with Ant System.

1 Introduction

Ant algorithms represent a relatively new heuristic search technique that has been successfully applied to solving NP hard problems [1]. Perhaps not surprisingly ant algorithms are biologically inspired from the behavior of colonies of real ants, and in particular how they forage for food. One of the main ideas behind this approach is that the ants can communicate with one another wholly through indirect means by making modifications to the pheromone level in their immediate environment.

Ant Colony Optimization (ACO) is the so-called meta-heuristic for ant algorithms applied to optimization problems and as these are the problems we're generally working with we tend to use the terms interchangeably.

The Traveling Salesman Problem (TSP) is an NP complete problem addressed by the optimization community having been studied extensively and been the target of considerable research [7]. The TSP is recognized as an easily understood, hard optimization problem of finding the shortest circuit of a set of cities starting from one city, visiting each other city exactly once, and returning to the start city again. Formally, the TSP is the problem of finding the shortest Hamiltonian circuit of a set of nodes (or in this case cities). There are two classes of TSP problem: symmetric TSP, and asymmetric TSP (ATSP). The difference is that whereas with symmetric

TSP the distance between two cities is the same regardless of the direction you travel, with asymmetric TSP this is not necessarily the case.

Ant Colony Optimization has been successfully applied to both classes of TSP with good results. The ACO algorithm skeleton for TSP is as follows [7]:

procedure ACO algorithm for TSPs Set parameters, initialize pheromone trails while (termination condition not met) do ConstructSolutions ApplyLocalSearch % optional UpdateTrails end end ACO algorithm for TSPs

The earliest implementation, Ant System, was just applied to the symmetric TSP problem initially and as this paper presents a proposed improvement to Ant System this is where we will focus our efforts.

While the ant foraging behaviour on which the Ant System is based has no central control or global information on which to draw, the use of global best information in the Elitest form of the Ant System represents a significant departure from the purely distributed nature of ant-based foraging. Use of global information presents a barrier to a fully distributed implementation of Ant System algorithms in a live network. This observation motivates the fully distributed algorithm – the Ant System Local Best Tour – described in this paper.

The remainder of this paper consists of 5 sections. The next section provides further detail of the algorithm introduced above. The Local Best Tour (LBT) algorithm is then introduced and the experimental setup for its evaluation described. An analysis section follows, and the paper concludes with an evaluation of the algorithm with proposals for further experimentation and improvement.

2 Ant System (AS)

As mentioned, Ant System was the earliest implementation of Ant Colony Optimization and was applied to the symmetric Traveling Salesman Problem. The implementation is built on top of the ACO algorithm skeleton shown above, and subsequently the terminology introduced in the skeleton is applicable.

A brief description of the algorithm follows. For a comprehensive description of the algorithm see 0, 0, 0 or 0.

Using Local Information To Guide Ant Based Search 3

2.1 Algorithm

Expanding upon the algorithm above, an ACO consists of two main sections: *initialization* and a *main loop*. The main loop runs for a user-defined number of iterations. These are described below:

Initialization

- Any Initial parameters are loaded.
- Each of the roads is set with an initial pheromone value.
- Each ant is individually placed on a random city.

Main loop begins

•

Construct Solution

• Each ant constructs a tour by successively applying the probabilistic choice function and randomly selecting a city it has not yet visited until each city has been visited exactly once.

$$p_{ij}^{k}(t) = \frac{\left[\tau_{ij}(t)\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}}{\sum_{l \in N_{i}^{k}} \left[\tau_{il}(t)\right]^{\alpha} \cdot \left[\eta_{il}\right]^{\beta}}$$

The probabilistic function, $p_{ij}^{k}(t)$, is designed to favour the selection of a road that has a high pheromone value, τ , and high visibility value, η , which is given by: $1/d_{ij}$, where d_{ij} is the distance to the city. The pheromone linearity, α , and visibility linearity, β , are parameters used to tune the relative importance of pheromone and road length in selecting the next city.

Apply Local Search

• Not used in Ant System, but is used in several variations of the TSP problem where 2-opt or 3-opt local optimizers 0 are used.

Best Tour check

• For each ant, calculate the length of the ant's tour and compare to the best tour's length. If there is an improvement update it.

Update Trails

- Evaporate a fixed proportion of the pheromone on each road.
- For each ant perform the "ant-cycle" pheromone update.
- Reinforce the best tour with a set number of "elitist ants" performing the "ant-cycle" pheromone update.

In the original investigation of Ant System algorithms, there were three versions that differed in how and when they laid pheromone. They are:

- "<u>Ant-density</u>" updates the pheromone on a road traveled with a fixed amount after every step.
- "<u>Ant-quantity</u>" updates the pheromone on a road traveled with an amount proportional to the inverse of the length of the road after every step.
- "<u>Ant-cycle</u>" first completed the tour and then updates each road used with an amount proportional to the inverse of the total length of the tour.

Of the three approaches "Ant-cycle" was found to produce the best results and subsequently receives the most attention. It will be used for the remainder of this paper.

Main Loop Ends

Output

• The best tour found is returned as the output of the problem.

2.2 Discussion

Ant System in general has been identified as having several good properties related to directed exploration of the problem space with out getting trapped in local minima 0. The initial form of AS did not make use of elitist ants and did not direct the search as well as it might. This observation was confirmed in our experimentation.

The addition of elitist ants was found to improve ant capabilities for finding better tours in fewer iterations of the algorithm, by highlighting the best tour. However, by using elitist ants to reinforce the best tour the problem is now taking advantage of global data and there is the additional problem of deciding on precisely how many elitist ants to use. If one uses too many elitist ants the simulation can easily become trapped in local minima 0, 0.

There have been a number of improvements to the original Ant System algorithm. They have generally focused on two main areas of improvement 0. First, they more strongly exploiting the globally best solution found. Second, they make use of a fast local search algorithm like 2-opt, 3-opt, or the Lin-Kernighan heuristic to improve the solutions found by the ants.

The algorithm improvements to Ant System have produced some of the highest quality solutions when applied to the TSP and other NP complete (or NP hard) problems 0.

As described in section 2.1, retrofitting AS with a local search facility would be straightforward. The area of improvement proposed in this paper is to explore an

alternative to using the globally best tour (GBT) to reinforce and focus on good areas of the search space. The Local Best Tour algorithm is described in the next section.

3 Local Best Tour (LBT)

The use of an elitist ant in Ant System exposes the need for a global observer to watch over the problem and identify what the best tour found to date is on a per iteration basis.

The idea behind the design of Local Best Tour is specifically to avoid this notion of a global observer from the problem. Instead, each individual ant keeps track of the best tour it has found to date and uses it in place of the elitist ant tour to reinforce tour goodness.

It is as if the scale of the problem has been brought down to the ant level and each ant is running its individual copy of the Ant System algorithm using a single elitist ant. Remarkably, the ants work together effectively even if indirectly and the net effect is very similar to that of using the pheromone search focusing of the elitist ant approach.

3.1 Algorithm

The algorithm used is identical to that described for Ant System with the replacement of the elitist ant step with the ant's local best tour step. Referring, once again, to the algorithm described in section 0, the following changes are made:

- That is, where the elitist ant step was:
 - Reinforce the best tour with a set number of "elitist ants" performing the "ant-cycle" pheromone update.

For Local Best Tour we now do the following:

• For each ant perform the "ant-cycle" pheromone update using its local best tour.

The rest of the Ant System algorithm is unchanged, including the newly explored tour's "ant-cycle" pheromone update.

3.2 Experimentation and Results

3.2.1 Overview

For the purposes of demonstrating Local Best Search we constructed an Ant System simulation and applied it to a series of TSP Problems from the TSPLIB95 collection 0. Although not the only problem studied – "eil76" and "kro101" were also

studied -- the data presented in this paper comes from running the simulation against the symmetric TSP problem "eil51", a problem for which the optimal tour is known. This problem is, perhaps not surprisingly, a 51-city problem set up in a 2 dimensional Euclidean plane. The weight assigned to each road comes from the linear distance separating each pair of cities.

The simulation created for this paper was able to emulate the behavior of the original Ant System, Ant System with elitist ants, and finally Ant System using the local best tour approach proposed in this paper. These three approaches were tested thoroughly to examine their relative performance on "eil51."

3.2.2 Parameters and Settings

Ant System requires you to make a number of parameter selections. These parameters are:

- Pheromone linearity
- Visibility linearity
- Pheromone decay rate
- Initial pheromone
- Pheromone additive constant
- Number of ants
- Number of elitist ants

In his original work on Ant System Marco Dorigo did work to tune and find appropriate values for a number of these parameters 0. The values Dorigo found that do not depend on the size of the problem were used for this simulation.

- Pheromone Linearity $(\alpha) = 1$
- Visibility Linearity $(\beta) = 5$
- Pheromone Decay Rate (ρ) = 0.5
- Initial Pheromone $(\tau_0) = 10^{-6}$

For those parameters that depend on the size of the problem our simulation made an effort to select good values based on knowledge of the problem and number of cities. Recent work 0 on improved algorithm parameters was unavailable to us when developing the LBT algorithm. We intend to explore the performance of the new parameters settings and will report the results in a future research paper.

The Pheromone additive constant (Q) was eliminated altogether as a parameter by replacing it with the global best tour (GBT) length in the case of standard Ant System and the local best tour (LBT) length for the approach in this paper. We justify this decision by noting that Dorigo found that differences in the value of Q only weakly affected the performance of the algorithm and a value within an order of magnitude of the optimal tour length was acceptable. This means that the pheromone addition on an edge becomes:

$$\frac{L_{best}}{L_{ant}}$$
 For a normal "ant-cycle" pheromone update

$$\frac{L_{best}}{L_{best}} = 1$$
 For an elitist or LBT "ant-cycle" pheromone update

The key factor in the pheromone update is that it remains inversely proportional to the length of the tour and this still holds with our approach. The ants now are not tied to a particular value of Q in the event of a change in the number of cities in the problem. We consider the removal of a user-defined parameter another attractive feature of the LBT algorithm.

For the number of ants (m) we simply set this equal to the number of cities, as this seems to be a reasonable selection according to the current literature 0, 0, 0.

For the number of elitist ants we tried various values dependant on the size of the problem and eventually settled on a value of $1/6^{th}$ of the number of cities. This value worked well for the relatively low number of cities we used in our simulation but for larger problems this value might need to be tuned. The current literature is unclear on the best value of the number of elitest ants to be used.

With the LBT AS approach, all ants perform the LBT "ant-cycle" update so subsequently the number of elitist ants is not needed. This paper does indicate the notion of an LBT constant that allows tuning of the LBT pheromone update rate as a possible improvement. This, however, this is not explored in the experimental results of this paper and subsequently the LBT constant is not varied and simply set to 1.

3.2.3 Results

Overall, the results of the simulation showed considerable promise for the Local Best Tour approach. Tables 1, 2, and 3 show a comparison of the results for the three Ant System approaches explored on the eil51 TSP problem after 100, 500, and 2000 iterations respectively. The results shown are the average best tour lengths and the overall best tour lengths after 100 runs.

| | Best Tour | Average | Std. Dev. | |
|-------------|-----------|---------|-----------|--|
| LBT AS | 437 | 442.4 | 5.23 | |
| Elitist AS | 430 | 443.4 | 6.77 | |
| Original AS | 443 | 451.5 | 4.67 | |

Table 1. Ant System Results for eil51 (100 iterations)

After 100 iterations we see that the LBT AS algorithm is producing a slightly better average result than the Elitist algorithm. On the other hand, the Elitist AS algorithm has over the course of the 100 runs been able to produce the best result. The standard deviations shown are all relatively high here however, the original AS algorithm despite showing relatively poor results has the lowest standard deviation.

| | Best Tour | Average | Std. Dev. | |
|-------------|-----------|---------|-----------|--|
| LBT AS | 430 | 433.4 | 2.91 | |
| Elitist AS | 429 | 438 | 4.29 | |
| Original AS | 438 | 445.1 | 5.99 | |

Table 2. Ant System Results for eil51 (500 iterations)

After 500 iterations the LBT AS algorithm has increased its lead for producing the best average results. Again, the Elitist AS produced the single best result although its best tour average was a fair bit worse than that of LBT AS. The standard deviation of the LBT AS is now substantially lower than the other two algorithms.

| | Best Tour | Average | Std. Dev. | |
|-------------|------------------|---------|-----------|--|
| LBT AS | 428 | 429.4 | 1.35 | |
| Elitist AS | 428 | 429.9 | 1.20 | |
| Original AS | 428 | 441.4 | 8.34 | |

Table 3. Ant System Results for eil51 (2000 iterations)

After 2000 iterations all of the algorithms found a best tour of length 428. The optimal tour length for this problem is actually 426 and so, unfortunately, none of the approaches were successful in finding it 0. This is however, in line with previous findings 0. That being said, the LBT Ant System was very effective and, in fact, had the best average results. Both the Elitist AS and LBT AS had relatively good, and really quite similar, results over the extended experiment. Although the original AS also tied the other approaches and managed to find a path of length 428, its average and standard deviation were both quite high relative to the other two Ant Systems.

Overall, the result for LBT AS show that it is at least in a similar class as the Elitist AS approach. The classical AS algorithm does not really manage to keep up and certainly requires many more iterations to achieve a similar performance for the final output best tour length.

4 Analysis

4.1 Best Tour Analysis

As has been shown in the Results section, LBT AS is competitive with the Elitist AS approach. In this section we take a comparative look at the evolution of the best tour in all three systems and then a look at the evolution of the best tour found per iteration.



Figure 1. Evolution of Best Tour Length

In Figure 1, we see can see the key difference between the Elitist AS approach and LBT AS. Whereas Elitist AS quickly finds a few good results, holds steady and then improves in relatively large pronounced steps, LBT AS improves more gradually at the beginning but continues its downward movement at a steadier rate. In fact, if one looks closely at the graph one can see that even the classical AS system has found a better result during the early stages of the simulation compared to LBT AS. However, by about iteration 75, LBT AS has overtaken the other two approaches and continues to gradually make improvements and maintains its overall improvement until the end of the simulation.

Overall, LBT AS's behavior could be described as slower but steadier. It takes slightly longer at the beginning to focus pheromone on good tours but after it has, it improves more frequently and steadily and on average will overtake the other two approaches given enough time. We have confirmed this hypothesis by experimentation with the eil76 and kro101 TSP problem datasets.



Figure 2. Best Tour Length for Individual Iterations

In Figure 2 we can see the distribution of best tours is more tightly focused with a smaller deviation for LBT AS compared to the other two systems. In addition, on average the LBT AS best tour space is much closer to the optimal solution.

As mentioned earlier in the paper none of the Ant Systems were able to find the optimal tour length for this problem. Judging from the flattening of the LBT AS graph line there looks to be a very real danger that the algorithm is caught in a local minimum. However, there is still a significant deviation in the best tour per iteration results. Later in this paper we look at ways to improve the algorithm and safeguard against this behaviour.

4.2 Average Tour Analysis

In the Best Tour Analysis we saw that there was a tendency for the LBT AS algorithm to gradually improve in many small steps. With our analysis of the average tour we want to confirm that the relatively high deviation of ant algorithms is working in the average case meaning that we are continuing to explore the problem space effectively. In this section we look at the average tour length per iteration to see if we can identify any behavioural trends.

In Figure 3 we see a very similar situation to that of the Best Tour Length per Iteration. The LBT AS algorithm is on average exploring much closer to the optimal solution. Perhaps more importantly, the LBT AS graph trend line is behaving very similarly in terms of its deviation as that with the other two systems. This suggests that the LBT AS system is working as expected and is in fact searching in a better-focused fashion closer to the optimal solution. Referring to Figure 2, one striking observation with LBT is the point in the experiment at which the best tour typically



emerges; 900 iterations versus well over a 1000 for the elitest ant algorithm and in excess of 1500 iterations for the basic AS algorithm.

Figure 3. Average Tour Length for Individual Iterations

4.3 Evolution of the Local Best Tour

The Local Best Tour approach is certainly very similar to the notion of elitist ants; only it is applied at the local level instead of at the global level. In this section we look at the evolution of the local best tour in terms of the average and worst tours, and compare them with the global best tour used by elitist ants.



Figure 4. Evolution of the Local Best Tour

From Figure 4 we can see that over time both the average and worst LBTs approach the value of global best tour. In fact the average in this simulation is virtually the

same as the global best tour. From this figure, it is clear that the longer the simulation runs the closer the LBT "ant-cycle" pheromone update becomes to that of an elitist ant's update scheme. However, as noted in the previous section, the number of iterations required by the LBT algorithm is considerably smaller than that required for the elitest ant algorithm.

5 Discussion and Conclusions

5.1 Capability of Local Best Tour

Through the results and analysis shown in this paper, Local Best Tour has proven to be a reasonable alternative to the use of the globally best tour for focusing ant search through pheromone reinforcement. In particular, the results show that the LBT Ant System algorithm has excellent average performance characteristics. By removing the need for the global information required for an elitest ant algorithm, we have improved the ease with which a parallel or live network implementation can be achieved; i.e. a completely distributed implementation of the TSP is possible.

Analysis of the best tour construction process shows that LBT AS, while initially converging more slowly than Elitist AS, is very consistent at incrementally building a better tour and on average will match or overtake the Elitist approach early in the search of the problem space.

Average and best iteration tour analysis has shown that LBT AS shares the same variability characteristics of the original Ant System that make it resistant to getting stuck in local minima. Furthermore, LBT AS is very effective in focusing its search towards the optimal solution.

Finally, LBT AS follows in the notion that the use of best tours to better focus an ant's search is an effect optimization. The emergent behaviour of a set of autonomous LBT ants is to, in effect, become elitist ants overtime.

5.2 Improvements and Future Work

As described earlier in this paper, a relatively straightforward way to further improve the performance of LBT AS would be to add a fast local search algorithm like 2-opt, 3-opt or the Lin Kernighan heuristic. This would dramatically improve the ability of the LBT AS algorithm to find better solutions earlier by squeezing the optimal solutions out of every proposed ant tour 0. Alternatively, the integration of recent network transformation algorithms 0 should prove useful as local search operators.

One of the problems highlighted in the best tour analysis was the possibility of getting trapped in a local minimum. The chances of this situation occurring could be reduced by the use of an LBT scaling factor. That is to say, in the LBT "ant-cycle" pheromone update adding an amount diminished by the scaling factor instead of always adding 1.

The Ant Colony Algorithm is not terribly efficient being of complexity $O(n^3 x$ the number of iterations). One of the primary contributors to this is the creation of the probability function from the list of all cities at each movement for the ant. A candidate list of constant or log(n) length would certainly help reduce the sensitivity of the algorithm to a large number of cities 0. Furthermore the integration of a local optimizer derived from recent advances in TSP algorithms 0 may well prove advantageous.

This implementation used the individual ants to store the Local Best tour information. In a real world problem this might be infeasible. An alternate approach might be to store the LBT information at the city or node instead. In this way the city would send out an LBT ant to reinforce the tour at every time-step.

Finally, future work should include the application of the LBT algorithm to other problems such as: the asymmetric TSP, the Quadratic Assignment Problem (QAP), the Vehicle Routing Problem (VRP) and other problems to which ACO has been applied 0.

5.3 Conclusions

This paper has demonstrated that an ACO algorithm using only local information can be applied to the TSP without compromising performance. While exhaustive experimentation has yet to be completed, the early results are promising and appear to support a hypothesis that LBT is competitive with, or superior to, AS with elitest ants. We believe that LBT with the improvements outlined in the previous section will further enhance our confidence in the hypothesis and look forward to reporting on these improvements in a future research paper.

References

- Bonabeau E., Dorigo M., and Theraulaz G. Swarm Intelligence From Natural to Artificial Systems. Oxford University Press, New York NY, 1999.
- Dorigo M. & L.M. Gambardella. Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Transactions on Evolutionary Computation, 1(1):53-66. <u>ftp://iridia.ulb.ac.be/pub/mdorigo/journals/IJ.16-TEC97.US.pdf</u>
- Dorigo M. & L.M. Gambardella. Ant Colonies for the Traveling Salesman Problem. BioSystems, 43:73-81. <u>ftp://iridia.ulb.ac.be/pub/mdorigo/journals/IJ.15-BIOSYS97.pdf</u>
- Dorigo M., V. Maniezzo & A. Colorni. The Ant System: Optimization by a Colony of Cooperating Agents. IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26(1):29-41. <u>ftp://iridia.ulb.ac.be/pub/mdorigo/journals/IJ.10-SMC96.pdf</u>

- Dumitrescu A. and Mitchell J., Approximation Algorithms for Geometric Optimization Problems," in the Proceedings of the Ninth Canadian Conference on Computational Geometry, Queen's University, Kingston, Canada, August 11-14, 1997, pp. 229–232.
- Pilat M. and White T., Using Genetic Algorithms to optimize ACS-TSP. In Proceedings of the 3rd International Workshop on Ant Algorithms, Brussels, Belgium, September 12-14 2002.
- Reinelt G. TSPLIB | A Traveling Salesman Problem Library. ORSA Journal on Computing, 3:376-384, 1991. <u>http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/</u>
- Stützle T. and Dorigo M. ACO Algorithms for the Traveling Salesman Problem. In K. Miettinen, M. Makela, P. Neittaanmaki, J. Periaux, editors, Evolutionary Algorithms in Engineering and Computer Science, Wiley, 1999. <u>ftp://iridia.ulb.ac.be/pub/mdorigo/bookchapters/BC.06-EuroGen99.pdf</u>

Appendix:

Comparing Ant Colony System (ACS) With Other Approaches To Solving TSP [3]

Comparison of ACS with the genetic algorithm (GA), evolutionary programming (EP), simulated annealing (SA), and the annealing-genetic algorithm (AG), a combination of genetic algorithm and simulated annealing (Lin, Kao and Hsu, 1993). We report the best integer tour length, the best real tour length (in parentheses) and the number of tours required to find the best integer tour length (in square brackets). Results using EP are from (Fogel, 1993) and those using GA are from (Bersini, Oury and Dorigo, 1995) for KroA100, and from (Whitley, Starkweather and Fuquay, 1989) for Oliver30, Eil50, and Eil75. Results using SA and AG are from (Lin, Kao and Hsu, 1993). Oliver30 is from (Oliver, Smith and Holland, 1987), Eil50, Eil75 are from (Eilon, Watson-Gandy and Christofides, 1969) and are included in TSPLIB2 with an additional city as Eil51.tsp and Eil76.tsp. KroA100 is also in TSPLIB. The best result for each problem is in boldface. It is interesting to note that the complexity of all the algorithms is order of *n*. *t* (where *n* is the number of cities and *t* is the number of tours generated). It is therefore clear that ACS and EP greatly outperform GA, SA, and AG.

| Problem name | ACS | GA | EP | SA | AG | Optimum |
|--------------------|-------------|-----------|-----------|-----------|----------|----------|
| Oliver30 | 420 | 421 | 420 | 424 | 420 | 420 |
| (30-city problem) | (423.74) | (N/A) | (423.74) | (N/A) | (N/A) | (423.74) |
| | [830] | [3,200] | [40,000] | [24,617] | [12,620] | |
| Eil50 | 425 | 428 | 426 | 443 | 436 | 425 |
| (50-city problem) | (427.96) | (N/A) | (427.86) | (N/A) | (N/A) | (N/A) |
| | [1,830] | [25,000] | [100,000] | [68,512] | [28,111] | |
| Eil75 | 535 | 545 | 542 | 580 | 561 | 535 |
| (75-city problem) | (542.31) | (N/A) | (549.18) | (N/A) | (N/A) | (N/A) |
| | [3,480] | [80,000] | [325,000] | [173,250] | [95,506] | |
| KroA100 | 21,282 | 21,761 | N/A | N/A | N/A | 21,282 |
| (100-city problem) | (21,285.44) | (N/A) | (N/A) | (N/A) | (N/A) | (N/A) |
| | [4,820] | [103,000] | [N/A] | [N/A] | [N/A] | |