



HAL
open science

Comparison of Three Neighbor Generation Structures by Simulated Annealing Method to Solve Quadratic Assignment Problem

Ali Jahed, Misagh Rahbari

► **To cite this version:**

Ali Jahed, Misagh Rahbari. Comparison of Three Neighbor Generation Structures by Simulated Annealing Method to Solve Quadratic Assignment Problem. 10th International Conference of Iranian Operations Research Society (ICIORS 2017), University of Mazandaran, May 2017, Babolsar, Iran. hal-01962309

HAL Id: hal-01962309

<https://hal.science/hal-01962309>

Submitted on 20 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Comparison of Three Neighbor Generation Structures by Simulated Annealing Method to Solve Quadratic Assignment Problem

Ali Jahed ^{1,*}, Misagh Rahbari ²

¹ Department of Industrial Engineering, Islamic Azad University South Tehran Branch, Tehran, Iran

Email: alijahed.ie@gmail.com

² Department of Industrial Engineering, Faculty of Engineering, Kharazmi University, Tehran, Iran

Email: misagh.rahbari@yahoo.com

Abstract

There are three neighbor generation structures such as swap, reversion, insertion, methods in the literature, for solving quadratic assignment linear programming problems. Swap is selecting two random positions in permutation encoding representation solution and swapping elements of these positions is the easiest and most widely used way of generating of neighbor solutions and reversion is selecting two random positions in permutation encoding representation solution and reversing the direction between two randomly chosen elements and insertion is Selecting two random positions in permutation encoding representation solution and with due attention to number of chosen elements, reversing the direction between two randomly chosen elements, these methods can be so deferent in finally result. We want to use these methods in one of algorithm like Simulated annealing (SA) and show it has deferent result with these methods.

Keywords: Quadratic assignment linear programming, Simulated annealing (SA), Three neighbor generation structures.

1. Introduction

The quadratic assignment problem (QAP) was introduced by Koopmans and Beckmann in 1957 as a mathematical model for the location of a set of indivisible economical activities [1]. It is NP-hard and is considered to be one of the most difficult problems to be solved optimally. Consider the problem of allocating a set of facilities to a set of locations, with the cost being a function of the distance and flow between the facilities, plus costs associated with a facility being placed at a certain location. Previous research based on the comparison between some algorithm with together, Pardalos [2] compared the performance of four algorithms including simulated annealing and tabu search and found that “all of these approaches have almost the same performance”. Paulli [3] compared simulated annealing and tabu search and found that “when CPU time is taken into consideration, simulated annealing is clearly preferable to tabu search”. On the other hand, [4] finds that “RTS (Reactive Tabu Search) needs less CPU time than SA to reach average results in the 1% [of the best known value] region”. In 1998, summarizing the situation, Cela [5] commented that “There is no general agreement concerning the comparison of the performance of simulated annealing approaches with that of tabu search approaches for the QAP”. There were some research findings on the performance comparison between TS and SA suggest that TS performs better than SA [5–6]. Battitand Tecchiolli [5] reported that TS performed better than SA in terms of CPU time needed to reach a solution quality that is 1% from the best known solutions. The instances considered, however, are rather small. Another direct comparison between SA and TS was done by Chiang and Chiang [7] where they have compared the performance of SA, TS, Probabilistic TS, and Hybrid TS for solving the Facility Layout Problem, formulated as a QAP. Their results show that their basic TS approach performs better than SA. Other researchers have compared the performance



of SA and TS with other algorithms. The earliest one is the comparison of SA, TS, Genetic Algorithms, Great Deluge Algorithm, and Record-to-Record Travel for solving the QAP that were conducted by Sinclair [6]. His results on the comparisons between TS and SA show that TS provided better solutions than SA in 28 out of 37 cases. Comparisons between SA and TS with a hybrid ant colony system with local search, HAS-QAP, were reported in [8]. The results suggest that the performance of TS is better than SA on most of the instances tested. A more recent paper by Arostegui et al. [5] reported the comparison among SA, TS and genetic algorithms on a specific QAP domain, the Facilities Location Problem. They compared the performance of these algorithms on several variants of the Facilities Location Problem. When comparing TS and SA, their results show that TS surpasses SA in most instances. In contrast to these results, in an early study Paulli [9] has reported that SA outperforms TS when the same computation time is spent by both algorithms. While this result may be attributed to the implementation of TS without the speed-ups proposed by Taillard [10], in a very recent study, Paul [11] shows that an SA algorithm with an instance dependent restart strategy may perform actually better than TS algorithms when high solution quality is required. However, it is unclear how this result generalizes to general, instance-independent restart mechanisms. In fact, in [11], the number of iterations is set in dependence of each instance and each considered solution quality bound.

Some research based on the new algorithm or extend some algorithm. such as, Ahmed [12] presented a new reformulation of the problem and developed a Lexi search Algorithm (LSA) to obtain exact optimal solution to this problem. He performed a comparative study to show the efficiency of the algorithm against an existing algorithm for some medium sized instances from the QAP library, QAPLIB (Burkard et al., [13]; Burkard, [14]) Forghani and Mohammadi [15] presented an integrated quadratic assignment and continuous facility layout problem. They obtained the arrangement of facilities within the departments through the QAP. They presented mathematical model as a mixed-integer programming (MIP) to minimize total material handling cost. In addition, they presented a heuristic method to solve the problem for large-scale problems and using several illustrative numerical examples, the performance of the model was examined. Tasgetiren et al.[16] presented some metaheuristics to solve QAP problems. Tseng and Liang [17] presented a hybrid metaheuristic for the quadratic assignment problem. Wang [18] applied Tabu search to solve QAP problem.

2. Describe model

$$(QAP) \min \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n c_{ijkl} x_{ijkl} \quad (1)$$

subject to

$$\sum_{i=1}^n x_{ij} = 1, j = 1, \dots, n, \quad (2)$$

$$\sum_{j=1}^n x_{ij} = 1, i = 1, \dots, n, \quad (3)$$

$$x_{ij} = 0, 1, i = 1, \dots, n, j = 1, \dots, n. \quad (4)$$

Eq. (1) states the objective function of the QAP problem and it minimizes sum of costs associated with facility assignment. Eq. (2) ensures that only any facility is assigned to only one place. Eq. (3) specifies that only each location is considered only for one place. Eq. (4) states that all variables are binary.



3. Definition of algorithms

3.1 Simulated annealing algorithm

Simulated annealing (SA) is a generic probabilistic meta-heuristics for combinatorial optimization problem of locating a good approximation to the global optimum of a given function in a relatively large search space. During the search, SA not only accepts better solutions but also the worse solutions but with a decreasing probability. At higher temperatures, the probability of accepting worse solutions is much higher. But, as the temperature decreases, the probability of accepting worse solution decreases. The probability of acceptance is assigned the value:

$$\text{Probability of acceptance} = \exp^{(-\Delta Z / T)}$$

A randomly generated number is used to test whether the move is accepted. Finally to decreases and update temperature, we multiply the current temperature by a constant ALPHA:

$$\text{Update temperature} = T \times \alpha$$

The SA consists of two loops. The inner loop runs till maximum number of acceptance or study neighbors for current temperature reached. The outer loop check for the stopping condition to be met. Each time the inner loop is finished, the temperature is updated using an update temperature formula till primary temperature to become equal freezing temperature.

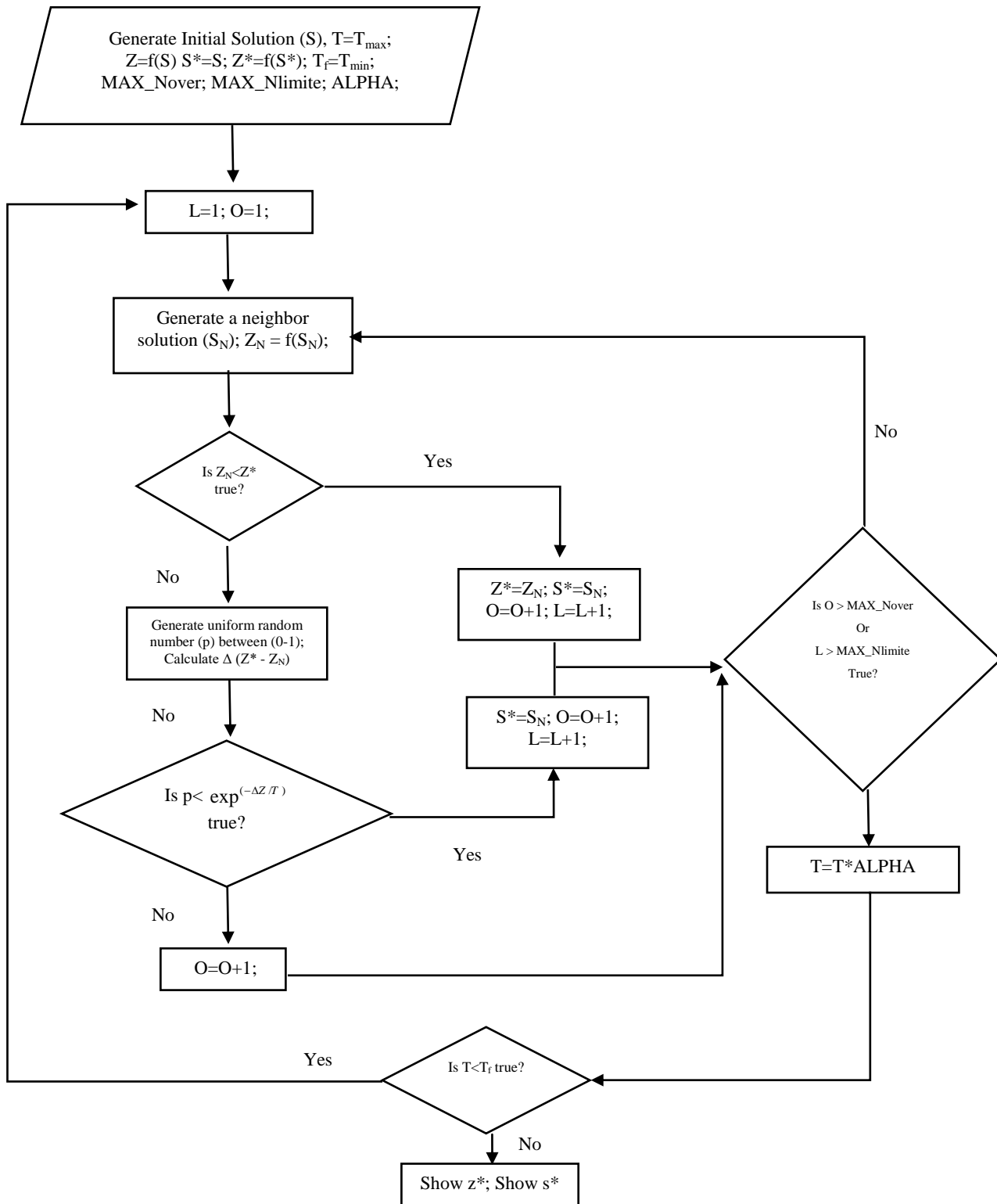


Figure 1. Algorithm of Simulated Annealing

3.2 Three neighbor generation structures

3.2.1 SWAP

Selecting two random positions in permutation encoding representation solution and swapping elements of these positions is the easiest and most widely used way of generating of neighbor solutions. In order to help to understand, in the **Figure 2**, we introduce the swap method. For example, if the third and fifth element be selected then their position will be replace by each other.

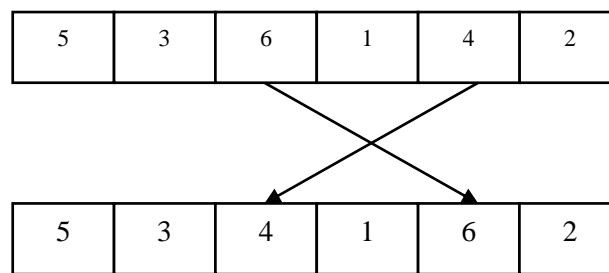


Figure 2. Neighbor Generation with Swap Operator

3.2.2 REVERSION

Selecting two random positions in permutation encoding representation solution and reversing the direction between two randomly chosen elements. In order to help to understand, in the **Figure 3**, we introduce the reversion method. For example, if the second and fifth element be selected then reversing the direction between their position.

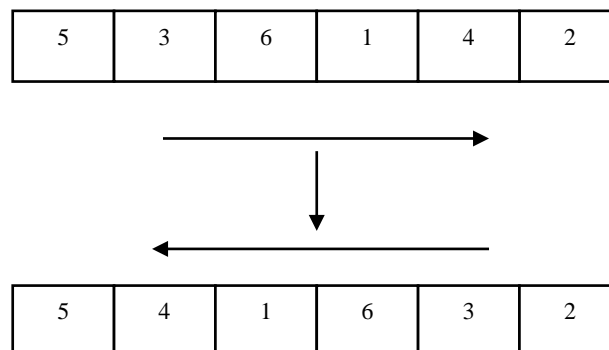


Figure 3. Neighbor Generation with Reversion Operator

3.2.3 INSERTION

Selecting two random positions in permutation encoding representation solution and with due attention to number of chosen elements, reversing the direction between two randomly chosen elements. In order to help to understand, in the **Figure 4**, we introduce the reversion method. For example, if the second and fifth element be selected then reversing the direction between their position.

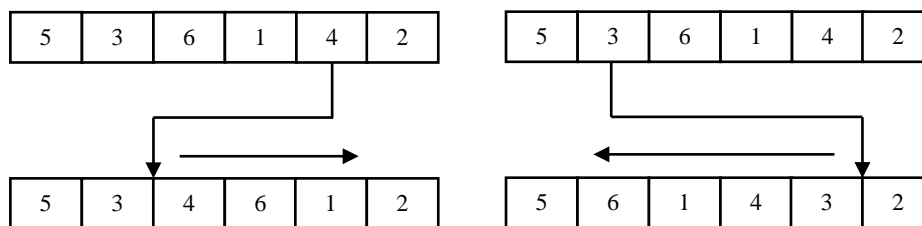


Figure 4. Neighbor Generation with Insertion Operator in 2 State

4. Computational results

In this section, we present details of the implementation of the proposed SA and compares the results with three neighbor generation structures. For the testing of these techniques, 3 example problems from the literature were used. All these techniques were written in MATLAB programming language and every technique was run 50 times for every problem. For the comparison of the techniques, GAP of the average results with due attention to the optimal solution was used.

$$GAP\% = \frac{average(Z) - optimal(Z)}{optimal(Z)} \times 100$$

Table 1 shows details of our results.

Name of neighbor generation	Problem number	Name of instances	n	Average Z	Optimal Z	GAP (%)
SWAP	1	Nug12	12	587.28	578	1.6
	2	Tai25(a)	25	1225460.4	1167256	4.98
	3	Tai50(a)	50	5195379.2	4938796	5.19
REVERSION	1	Nug12	12	596.4	578	3.18
	2	Tai25(a)	25	1289886.68	1167256	10.5
	3	Tai50(a)	50	5555950.08	4938796	12.49
INSERTION	1	Nug12	12	805.2	578	39.3
	2	Tai25(a)	25	1445281.12	1167256	23.8
	3	Tai50(a)	50	5904583.28	4938796	19.5

Table 1. The Summary of Comparison of Three Neighbor Generation Structures with SA

5. Conclusion

In this study, first we have presented quadratic assignment linear programming then we introduce three neighbor generation structure that used in proposed simulation annealing (SA) meta-heuristic called swap, insertion and reversion. We have presented a based on comparing of three methods to solve quadratic assignment linear programming. The technique were tested on 3well known benchmark problem sets from literature. The proposed SA based on insertion approach in this study we conclude that rate of gap in insertion approach totally is more than other approach but when the size of problem increased, the gap of problem decreased. The time of solve in insertion approach is better than other approach has been illustrated.



Acknowledgments

The first author greatly appreciates to the Research Deputy of University of Mazandaran for their support.

References

- [1] T. C. Koopmans and M. J. Beckmann, Assignment problems and the location of economic activities, *Econometrica* 25, 1957, 53–76.
- [2] P.M. Pardalos, K.A. Murthy, T.P. Harrison, A computational comparison of local search heuristics for solving local quadratic assignment problems, *Informatica* 4 (1993) 172–187.
- [3] J. Paulli, A computational comparison of simulated annealing and tabu search applied to the quadratic assignment problem, in: V. Ren, V. Vidal (Eds.), *Applied Simulated Annealing*, in: *Lecture Notes in Economics and Mathematical Systems*, Springer-Verlag, 1993, pp. 85–102.
- [4] E. Çela, *The Quadratic Assignment Problem: Theory and Algorithms*, Kluwer, Boston, MA, 1998.
- [5] Arostegui MA, Kadipasaoglu S, Khumawala BM. An empirical comparison of tabu search, simulated annealing, and genetic algorithms for facilities location problems. *Int J Prod Econ* 2006;103(2):742–54.
- [6] Battiti R, Tecchiolli G. Simulated annealing and tabu search in the long run: a comparison on QAP tasks. *Comput Math Appl* 1994;28(6):1–8. [6] Sinclair M. Comparison of the performance of modern heuristics for combinatorial optimization on real data. *Comput Oper Res* 1993;20(7):687–95, [http://dx.doi.org/10.1016/0305-0548\(93\)90056-O](http://dx.doi.org/10.1016/0305-0548(93)90056-O).
- [6] R. Battiti, G. Tecchiolli, Tabu search in the long-run—a comparison on QAP tasks, *Comput. Math. Appl.* 28 (1994) 1–8.
- [7] Chiang W-C, Chiang C. Intelligent local search strategies for solving facility layout problems with the quadratic assignment problem formulation. *Eur J Oper Res* 1998;106(2–3):457–88.
- [8] Gambardella LM, Taillard ÉD, Dorigo M. Ant colonies for the quadratic assignment problem. *J Oper Res Soc* 1999;50(2):167–76.
- [9] Paulli J. A computational comparison of simulated annealing and tabu search applied to the assignment problem. *Appl Simulated Annealing* 1993;46 (1):85–102.
- [10] Taillard ÉD. Robust taboo search for the quadratic assignment problem. *Parallel Comput* 1991;17(4–5):443–55.
- [11] Paul G. Comparative performance of tabu search and simulated annealing heuristics for the quadratic assignment problem. *Oper Res Lett* 2010;38 (6):577–81.
- [12] Ahmed, Z. H. (2013). A New Reformulation and an Exact Algorithm for the Quadratic Assignment Problem. *Indian Journal of Science & Technology*, 6(4), 4368-4377.
- [13] Burkard, R. E., Karisch, S. E., & Rendl, F. (1997). QAPLIB—a quadratic assignment problem library. *Journal of Global Optimization*, 10(4), 391-403.
- [14] Burkard, R. E. (2013). *Quadratic assignment problems* (pp. 2741-2814). Springer New York. Çela, E. (1998). *The Quadratic Assignment Problem: Theory and Algorithms*. Kluwer Academic Publishers, Dordrecht.
- [15] Forghani, K., & Mohammadi, M. (2012). Integrated quadratic assignment and continuous facility layout problem. *International Journal of Industrial Engineering Computations*, 3(5), 787-806.
- [16] Tasgetiren, M. F., Pan, Q. K., Suganthan, P. N., & Dizbay, I. E. (2013, April). Metaheuristic algorithms for the quadratic assignment problem. In *Computational Intelligence In Production And Logistics Systems (CIPLS), 2013 IEEE Workshop on* (pp. 131-137). IEEE.
- [17] Tseng, L. Y., & Liang, S. C. (2006). A hybrid metaheuristic for the quadratic assignment problem. *Computational Optimization and Applications*, 34(1), 85-113.
- [18] Wang, J. C. (2007, November). Solving quadratic assignment problems by a tabu based simulated annealing algorithm. In *Intelligent and Advanced Systems, 2007. ICIAS 2007. International Conference on* (pp. 75-80). IEEE.