Assessing Simulated Annealing with Variable Neighborhoods

Eduardo Lalla-Ruiz[1(✉)], Leonard Heilig[2], and Stefan Voß[2]

[1] Department of Business Information Systems and Industrial Engineering, University of Twente, Enschede, The Netherlands
e.a.lalla@utwente.nl
[2] Institute of Information Systems, University of Hamburg, Hamburg, Germany
{leonard.heilig,stefan.voss}@uni-hamburg.de

Abstract. Simulated annealing (SA) is a well-known metaheuristic commonly used to solve a great variety of $\mathcal{NP}$-hard problems such as the quadratic assignment problem (QAP). As commonly known, the choice and size of neighborhoods can have a considerable impact on the performance of SA. In this work, we investigate and propose a SA variant that considers variable neighborhood structures driven by the state of the search. In the computational experiments, we assess the contribution of this SA variant in comparison with the state-of-the-art SA for the QAP applied to printed circuit boards and conclude that our approach is able to report better solutions by means of short computational times.

1 Introduction

This paper proposes and assesses the incorporation of variable neighborhoods into SA driven by the state of the search. From a practical standpoint, the basic idea for this strategy was initially proposed in [7] for a vehicle routing application in order to obtain better solutions than those provided by a standard SA. At a methodological level, the contribution of our paper is to evaluate this strategy in a more general setting with standard problem instances.

In the quest of showing the impact of the algorithmic enhancement regarding a known method, i.e., SA, we exemplify by means of the quadratic assignment problem (QAP). The QAP is an $\mathcal{NP}$-hard combinatorial optimization problem introduced by Koopmans and Beckman [10] that have received a lot of attention due to its numerous applications. In the QAP, we are given a set of facilities denoted as $\mathcal{F} = \{1, 2, \ldots, n\}$ and a set of locations denoted as $\mathcal{L} = \{1, 2, \ldots, n\}$. Each pair of facilities, $(i, j) \in \mathcal{F}$, requires a certain flow, i.e., $f_{ij} \geq 0$. The distance between the locations $k, l \in \mathcal{L}$ is denoted as $d_{kl} \geq 0$. It should be mentioned that the flows and distances are symmetric (*i.e.*, $f_{ij} = f_{ji}, \forall i, j \in \mathcal{F}$ and $d_{kl} = d_{lk}, \forall k, l \in \mathcal{L}$) and the flow/distance between a given facility/location and itself is zero (*i.e.*, $f_{ii} = 0, \forall i \in \mathcal{F}$ and $d_{kk} = 0, \forall k \in \mathcal{L}$). Its objective is to minimize the cost derived from the distance and flows among facilities. Duman et al. [5] present a practical application of the QAP for sequencing placement

N. F. Matsatsinis et al. (Eds.): LION 13 2019, LNCS 11968, pp. 298–303, 2020.
https://doi.org/10.1007/978-3-030-38629-0_24

and configuration printed circuit boards (PCB) and extensively analyze the use of SA for addressing the problem. The QAP is formally expressed as follows

$$minimize \sum_{i=1}^{n} \sum_{j=1}^{n} f_{ij} d_{\phi(i)\phi(j)}, \tag{1}$$

where ϕ is a solution belonging to the set composed of all the feasible permutations, denoted as $\mathcal{S}_n$, such that $\phi : \mathcal{F} \rightarrow \mathcal{L}$. The cost associated to assign facility i to location $\phi(i)$ and facility j to facility $\phi(j)$ is, according to Eq. (1), $f_{ij} d_{\phi(i)\phi(j)}$. In addition, let us denote as $f(\phi)$ the objective function value of solution $\phi \in \mathcal{S}_n$. Drezner *et al.* [4] review the applicability of widespread metaheuristics from the literature to address the QAP. The interested reader is referred to the detailed survey provided by Loiola et al. [13].

We use the same problem instances of [6] and the best state-of-the-art SA [5] to properly compare our proposed approach. From the computational experiments, we conclude that our approach provides a better performance compared to the SA when using a single neighborhood structure.

The remainder of this paper is structured as follows. First, we review some related works in Sect. 2. The proposed variable SA algorithm is presented in Sect. 3. In Sect. 4, we report the results of the computational experiments. The paper ends with some conclusions and an outlook.

2 Related Works

In general terms, Cheh et al. [2] studied the effect that neighborhood structures have on SA. On the other hand, Ogbu and Smith [14] showed the benefit of using larger neighborhoods within SA. Henderson et al. [9] review the impact that the choice of neighborhoods has on SA and indicate that the efficiency of SA is highly influenced by the neighborhood selection.

Besides Heilig et al. [7], some authors have investigated ideas to control the neighborhood structure during the search. Xu and Qu [17] investigate the use of variable neighborhoods within an evolutionary multi-objective SA (EMOSA) for solving multicast routing problems. Using multiple neighborhood structures specifically designed for each objective significantly improves the performance of the SA. Ying et al. [19] propose an SA algorithm with variable neighborhoods and define additional parameters to control the random selection of the neighborhood structure. While the performance of the algorithm depends on a good configuration of the parameters, requiring additional experiments. The proposed approach was able to find new best-known solutions for the cell formation problem.

Instead of defining a parameter, Rodriguez-Cristerna and Torres-Jimenez [15] use only two neighborhood structures and select them with uniform probabilities. Some authors investigate ideas to adjust the size of neighborhood structures, such as by means of a non-uniform mutation operator for monotonously decreasing the neighborhood size [16] or by using a circle-directed mutation as done in [12].

Other than in previous works, we propose a dynamic neighborhood variation where the neighborhood structures are changed depending on the success of finding better solutions.

Furthermore, SA is used in some works to extend the acceptance criterion of the variable neighborhood search (VNS) for accepting also non-improving solutions under certain conditions (see, e.g., [3,8,11,18]).

3 Variable Neighborhood Simulated Annealing Algorithm

In order to evaluate the contribution of variable neighborhood within SA, the best state-of-the-art SA proposed for solving the QAP-PCB [5] is used as a base template. For extending it, we include the novel incorporation of neighborhood variation in lines 5 to 14, where a parameter k is introduced for regulating the change of neighborhood structures.

Algorithm 1. SA with variable neighborhood structures (SA-VN)

Require: $Temp_{min}$, α, β, r_{max}
1: $S \leftarrow$ generate initial solution at random
2: $Temp \leftarrow f_{obj}(S)\alpha$; $k \leftarrow 1$
3: **while** ($Temp_{min} < Temp$ and $it \leq it_{max}$) **do**
4:   **for** ($r = 1$ to r_{max}) **do**
5:     Generate a solution $S' \in \mathcal{N}^k(\phi)(S)$
6:     Calculate $\Delta_{S,S'} = f_{obj}(S') - f_{obj}(S)$
7:     **if** ($\Delta_{S,S'} \leq 0$) **then**
8:       $S \leftarrow S'$
9:       $k \leftarrow 1$
10:       Update best solution S_{best} if applicable
11:     **else**
12:       $S \leftarrow S'$ with probability $e^{-\Delta/Temp}$
13:       $k++$
14:     **end if**
15:     $Temp = Temp \cdot \beta$
16:     $it++$
17:   **end for**
18: **end while**
19: Return S_{best}

We apply the *swap* neighborhood structure. That is, given a solution, $\phi \in \mathcal{S}_n$, the swap neighborhood, $\mathcal{N}^1(\phi) = \{\phi \circ (i,j) : 1 \leq i,j \leq n, i \neq j\}$, performs the transposition (i,j) by swapping the two relevant locations assigned to the indexes, i and j, respectively. Moreover, we define $\mathcal{N}^2(\phi)$ and $\mathcal{N}^3(\phi)$ as the application of swap consecutively two and three times, respectively.

4 Computational Results

The computational experiments were executed over the instances proposed by Duman *et al.* [6] which were generated considering the QAP-PCB. In each case, 30 executions of our SA-VN were carried out for each problem instance. The algorithms were executed on a computer equipped with an Intel i7-7700HQ and 16 GB of RAM. In order to properly evaluate the contribution of our approach, we have used the SA and its best parameters as provided in [5] with a maximum number of $|n|^{3.5}$ solutions.

Table 1. Comparison among SA-VN algorithms for the QAP-PCB instances. Best values in bold.

	SA-VN (k = 3)				SA-VN (k = 2)			
Instance (size)	Min	Avg.	Max.	t (ms)	Min	Avg.	Max.	t (ms)
B1 (58)	**1066**	1084.33	1132	3281.47	**1066**	1088.33	1126	3302.47
B2 (54)	**754**	771.067	798	2865.37	756	775.4	796	2852.4
B3 (52)	**730**	749.667	784	2666.57	732	749.867	776	2653.37
B4 (50)	**1450**	1456	1482	1651.27	**1450**	1466.13	1502	1643.07
B5 (48)	**752**	765.133	802	1534.33	754	767.067	814	1525.43
B6 (49)	**1388**	1400.33	1436	1601.3	1392	1397.07	1438	1597.5
B7 (47)	1350	1362.6	1394	1474.2	**1348**	1360.6	1374	1503.83
B8 (40)	**714**	725.067	740	491.533	718	726.333	738	497.667
	1025.5	1039.28	1071	1945.75	1027	1041.35	1070.5	1946.97

Table 2. Comparison among SA algorithms using one neighborhood structure for the QAP-PCB instances. Best values in bold.

	SA_1				SA_2				SA_3			
Instance (size)	Min	Avg.	Max.	t (ms)	Min	Avg.	Max.	t (ms)	Min	Avg.	Max.	t (ms)
B1(58)	1070	1087.73	1130	3319	**1066**	1084.13	1126	3323.57	1068	1092	1130	3345.23
B2 (54)	758	778.4	814	2850.1	756	775.533	814	2894.47	760	774.60	796	2938.67
B3 (52)	**730**	753.4	780	2667.2	734	757.733	784	2701.37	**730**	752.67	804	2744.90
B4 (50)	**1450**	1458.4	1488	1628.9	**1450**	1462.47	1484	1664.33	**1450**	1463.73	1514	1696.17
B5 (48)	**752**	764.533	804	1512.07	754	766.533	802	1533.57	754	764.20	808	1556.43
B6 (49)	1390	1399.93	1438	1589.63	**1388**	1397.53	1444	1589.5	1390	1399.20	1438	1638.93
B7 (47)	1350	1362.07	1386	1454.9	**1348**	1360	1378	1472.3	**1348**	1364.40	1398	1512.27
B8 (40)	718	724.867	736	487.1	716	724.667	734	492.133	716	725.20	736	505.87
	1027.25	1041.17	1072	1938.61	1026.5	1041.08	1070.75	1958.9	1027	1042	1078	1992.31

Table 1 shows the comparison between SA with variable neighborhoods considering two, i.e., SA-VN (k = 2) and three neighborhoods, i.e., SA-VN (k = 3).

Moreover, the best SA proposed by [5] considering individually all the used neighborhoods is compared in Table 2. For each problem instance, the performances of the algorithms in terms of average objective value (Avg.), best objective value (Min.), the worst objective value (Max.), and the computational time (t (ms)) of all the executions in milliseconds are reported.

From the results, it can be seen that all algorithms require similar computational times. The strategy of including variable neighborhoods permits to obtain more best-known solutions. Although there is not a relevant difference in terms of the worst values, the average performance is enhanced when more neighborhoods are considered in SA-VN. Moreover, there is a relevant performance benefit when the number of neighborhoods increases.

5 Conclusions

In this work, a novel simulated annealing with variable neighborhoods changing along the search is proposed for solving the quadratic assignment problem. This new SA approach includes alternating neighborhoods when there is no improvement or the probability of acceptance does not permit a worsening movement. It is noticeable from the numerical experiments that the proposed algorithm exhibits a better performance within similar time frames as the standard SA. The promising results encourage to further explore this research direction. The results also go in line with those shown in [7] where the inclusion of variable neighborhoods leads to an overall improvement of the SA search framework.

As future work, we aim at extending and analyzing the performance of SA-VN on other QAP instances such as those from the QAPLIB [1] as well as other optimization problems. Moreover, we aim to add a look ahead component like known from the pilot method.

References

1. Burkard, R.E., Karisch, S.E., Rendl, F.: QAPLIB-a quadratic assignment problem library. J. Global Optim. **10**(4), 391–403 (1997)
2. Cheh, K.M., Goldberg, J.B., Askin, R.G.: A note on the effect of neighborhood structure in simulated annealing. Comput. Oper. Res. **18**(6), 537–547 (1991)
3. Chen, P., Huang, H.K., Dong, X.Y.: Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem. Expert Syst. Appl. **37**(2), 1620–1627 (2010)
4. Drezner, Z., Hahn, P.M., Taillard, E.D.: Recent advances for the quadratic assignment problem with special emphasis on instances that are difficult for meta-heuristic methods. Ann. Oper. Res. **139**(1), 65–94 (2005)
5. Duman, E., Or, I.: The quadratic assignment problem in the context of the printed circuit board assembly process. Comput. Oper. Res. **34**(1), 163–179 (2007)
6. Duman, E., Uysal, M., Alkaya, A.F.: Migrating birds optimization: a new meta-heuristic approach and its performance on quadratic assignment problem. Inf. Sci. **217**, 65–77 (2012)

7. Heilig, L., Lalla-Ruiz, E., Voß, S.: port-IO: an integrative mobile cloud platform for real-time inter-terminal truck routing optimization. Flex. Serv. Manuf. J. **29**(3), 504–534 (2017)
8. Hemmelmayr, V.C., Doerner, K.F., Hartl, R.F.: A variable neighborhood search heuristic for periodic routing problems. Eur. J. Oper. Res. **195**(3), 791–802 (2009)
9. Henderson, D., Jacobson, S.H., Johnson, A.W.: The theory and practice of simulated annealing. In: Glover, F.W., Kochenberger, G.A. (eds.) Handbook of Metaheuristics, pp. 287–319. Springer, Boston (2003). https://doi.org/10.1007/0-306-48056-5_10
10. Koopmans, T.C., Beckman, M.: Assignment problems and the location of economic activities. Econometrica **25**, 53–76 (1957)
11. Kuo, Y., Wang, C.C.: A variable neighborhood search for the multi-depot vehicle routing problem with loading cost. Expert Syst. Appl. **39**(8), 6949–6954 (2012)
12. Lin, Y., Bian, Z., Liu, X.: Developing a dynamic neighborhood structure for an adaptive hybrid simulated annealing-tabu search algorithm to solve the symmetrical traveling salesman problem. Appl. Soft Comput. **49**, 937–952 (2016)
13. Loiola, E.M., Maia de Abreu, N.M., Boaventura-Netto, P.O., Hahn, P., Querido, T.: A survey for the quadratic assignment problem. Eur. J. Oper. Res. **176**(2), 657–690 (2007)
14. Ogbu, F., Smith, D.K.: The application of the simulated annealing algorithm to the solution of the n/m/Cmax flowshop problem. Comput. Oper. Res. **17**(3), 243–253 (1990)
15. Rodriguez-Cristerna, A., Torres-Jimenez, J.: A simulated annealing with variable neighborhood search approach to construct mixed covering arrays. Electron. Notes Discret. Math. **39**, 249–256 (2012)
16. Xinchao, Z.: Simulated annealing algorithm with adaptive neighborhood. Appl. Soft Comput. **11**(2), 1827–1836 (2011)
17. Xu, Y., Qu, R.: Solving multi-objective multicast routing problems by evolutionary multi-objective simulated annealing algorithms with variable neighbourhoods. J. Oper. Res. Soc. **62**(2), 313–325 (2011)
18. Xu, Y., Wang, L., Yang, Y.: A new variable neighborhood search algorithm for the multi depot heterogeneous vehicle routing problem with time windows. Electron. Notes Discret. Math. **39**, 289–296 (2012)
19. Ying, K.C., Lin, S.W., Lu, C.C.: Cell formation using a simulated annealing algorithm with variable neighbourhood. Eur. J. Ind. Eng. **5**(1), 22–42 (2010)