# Exploring the Behaviour of the Descent-Ascent Heuristic Optimization Algorithm

Prof. Alfonsas Misevičius, Assoc. Prof. Gintaras Palubeckis, Assoc. Prof. Dovilė Verenė
Faculty of Informatics, Kaunas university of technology

ktu 1922

## INTRODUCTION

— The essence of the proposed descent-ascent (DA) algorithm lies in the general principle that progress may (often) require temporary setbacks (failures) (Fig. 1)

— In the context of heuristic algorithms, this implies that effective performance may be attained by combining solution improvements (i.e., descents, in the context of minimization problems) with some perturbations (mutations, “invasions” (i.e., ascents)

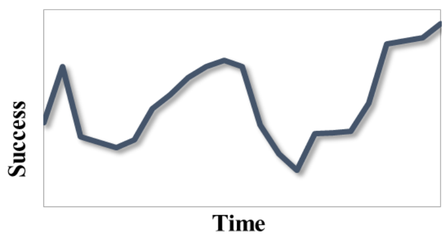


Fig 1. Success vs. time

## THE DA ALGORITHM. PSEUDOCODE

**Algorithm 1** Top-level pseudocode of the descent-ascent principle based algorithm

```
Descent-Ascent_Principle_Based_Algorithm;
// input: n – the problem size, A, B – data matrices,
//        DA_Iter_N – total number of iterations of the DA algorithm,
//        LS_Iter_N – the number of iterations of the local search algorithm (descent procedure),
//        RP_Iter_N_Fact – random perturbation factor (ascent factor),
//        Q1, Q2 – the user-defined parameters (integers) (lower and higher (prime) numbers) (Q1<Q2 – 1)
// output: p* – the best found solution
01 begin
02   create the initial solution p;
03   i ← 1;
04   j ← Q1;
05   if j is the prime number then
06        perform random move with respect to p
07   else begin // j is not a prime number
08     if the previous number (j – 1) is the prime number then begin
09        try to improve the current solution p (if it is possible);
10        if the solution is improved then goto 35 else goto 36
11     endthen
12     else begin // j – 1 is not a prime number
13       if there was improvement of p in the previous iteration
14           then begin
15           try to improve the current solution p (if it is possible);
16           if the solution is improved then goto 35 else goto 36
17       endthen
18       else begin // there was no improvement in the previous iteration
19         if Random_Number(0, 1) < 1/3 then begin
20            perform random move with respect to p;
21            try to improve the current solution p
22         endthen
23         else if Random_Number(0, 1) < 2/3 then begin
24                 perform increased random move with respect to p;
25                 try to improve the current solution p
26               endthen
27               else begin // Random_Number(0, 1) >= 2/3
28                 perform random move or increased random move
29                 with the equal probability (1/2) with respect to p;
30                 try to improve the current solution p
31               endelse
32       endelse // there was no improvement in the previous iteration
33     endelse // j – 1 is not a prime number
34   endelse; // j is not a prime number
35   if z(p) < z(p') then p' ← p; // the best so far solution is memorized
36   if j < Q2 then j ← j + 1 else j ← Q1; // go to the next number
37   i ← i + 1; // go to the next iteration
38   if i <= DA_Iter_N then goto 05 else goto 39;
39   return p'
40 end.
```

## THE EXPLORED COMPONENTS/PARAMETERS OF THE DA ALGORITHM

— ***N*** - the total number of iterations of the DA algorithm;

— ***T*** - the number of iterations of the tabu search (TS) procedure within the iterated tabu search algorithm;

— ***IT*** - the number of iterations of the iterated tabu search (ITS) algorithm within the DA algorithm itself;

— ***MN*** - the number of multistarts (restarts).

## CASE STUDY: THE QUADRATIC ASSIGNMENT PROBLEM (QAP)

— QAP: the goal is to find a (globally optimal) permutation p=(p(1), p(2),…,p(n)), such that the following objective function is minimized (among all permutations for given matrices A, B):

$$z(p) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij} b_{p(i)p(j)}$$

— The following benchmark instances have been used in the experiments: tai40a, tai50a, tai60a, tai80a, tai100a, tai27e1..tai27e5, tai45e1..tai45e5, tai75e1..tai75e5, tai125e1..tai125e5, tai175e1..tai175, bl49. They are from the acknowledged sources (QAPLIB - Quadratic Assignment Problem Library, / Burkard et. al, 1997/, /Drezner et. al, 2005/, Taillard – QAP (http://mistic.heig-vd.ch/taillard/problemes.dir/qap.dir/qap.html), /De Carvalho et al., 2006/)

## EMPIRICAL RESULTS

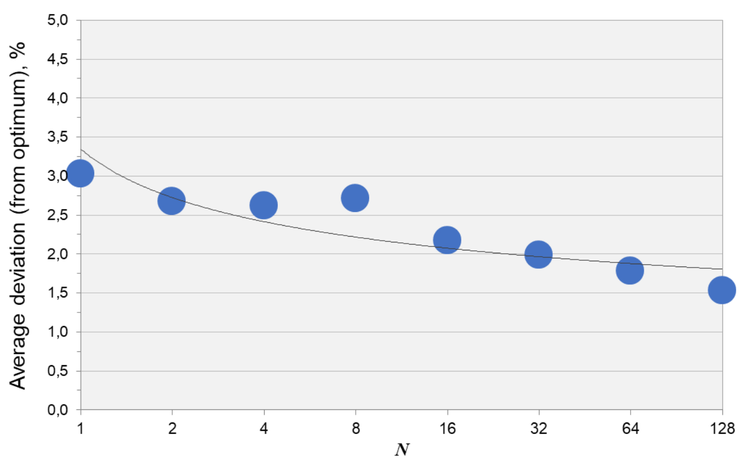


Fig 2. The average percentage deviations from optimum are given for the instance “bl49”. The x-axis of the graph represents the numbers of iterations (*N*) of the DA algorithm

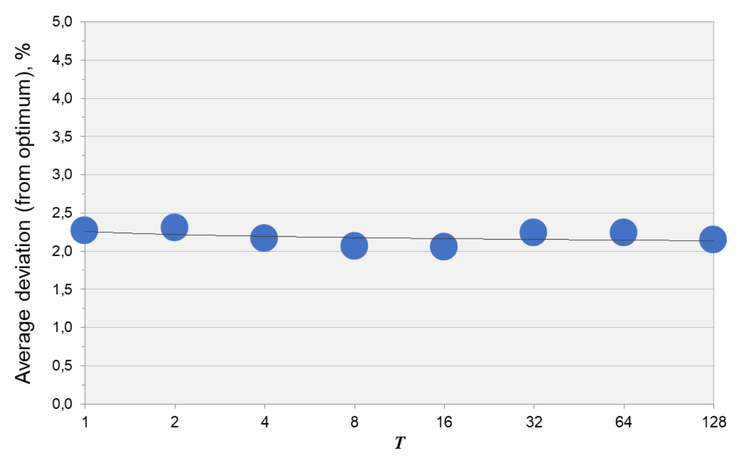


Fig 3. The average percentage deviations from optimum are given for the instance “bl49”. The x-axis represents the numbers of TS iterations (*T*)

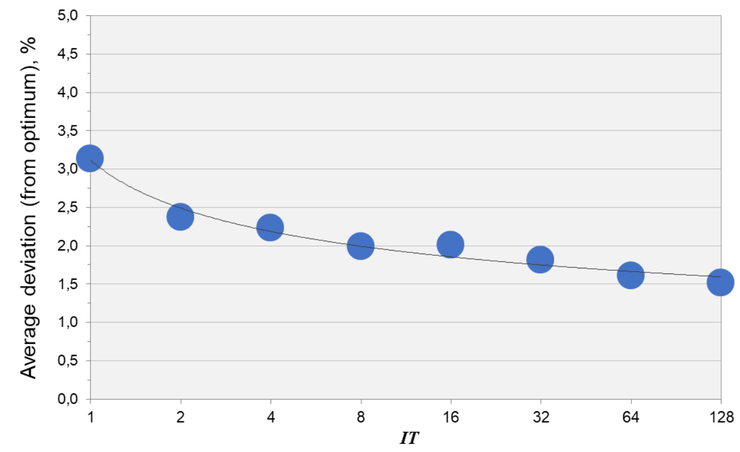


Fig. 4. The average percentage deviations from optimum are given for the instance “bl49”. The x-axis represents the numbers of ITS iterations (*IT*)

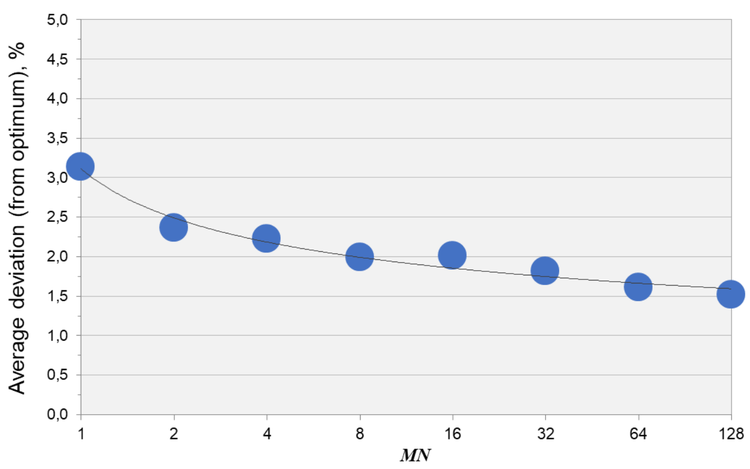


Fig 5. The average percentage deviations from optimum are given for the instance “bl49”. The x-axis represents the numbers of multistarts (*MN*)

## EMPIRICAL RESULTS. LONG RUNS

| Instance | Best known value | Deviation, % | Time (sec) |
|---|---|---|---|
| tai40a | 3139370 | 0.000 | 841.400 |
| tai50a | 4938796 | 0.000 | 2650.000 |
| tai60a | 7205962 | 0.000 | 8735.000 |
| tai80a | 13499184 | 0.121 | 67370.000 |
| tai100a | 21043560 | 0.145 | 97330.000 |
| tai27e1 | 2558 | 0.000 | 0.106 |
| tai27e2 | 2850 | 0.000 | 0.096 |
| tai27e3 | 3258 | 0.000 | 0.060 |
| tai27e4 | 2822 | 0.000 | 0.071 |
| tai27e5 | 3074 | 0.000 | 0.059 |
| tai45e1 | 6412 | 0.000 | 0.567 |
| tai45e2 | 5734 | 0.000 | 0.457 |
| tai45e3 | 7438 | 0.000 | 0.668 |
| tai45e4 | 6698 | 0.000 | 0.557 |
| tai45e5 | 7274 | 0.000 | 0.487 |
| tai75e1 | 14488 | 0.000 | 5.677 |
| tai75e2 | 14444 | 0.000 | 5.984 |
| tai75e3 | 14154 | 0.000 | 5.749 |
| tai75e4 | 13694 | 0.000 | 5.324 |
| tai75e5 | 12884 | 0.000 | 5.883 |
| tai125e1 | 35426 | 0.000 | 780.200 |
| tai125e2 | 36178 | 0.000 | 845.600 |
| tai125e3 | 30498 | 0.000 | 693.000 |
| tai125e4 | 33084 | 0.000 | 911.900 |
| tai125e5 | 37210 | 0.000 | 915.200 |
| tai175e1 | 57540 | 0.000 | 35440.000 |
| tai175e2 | 50110 | 0.000 | 34830.000 |
| tai175e3 | 53900 | 0.000 | 29950.000 |
| tai175e4 | 60416 | 0.000 | 30170.000 |
| tai175e5 | 50004 | 0.000 | 28550.000 |

## CONCLUDING REMARKS. RECOMMENDATIONS

— Overall, it can be stated that the introduced idea has demonstrated (quite) strong *potential* (and *effectiveness*).

— The proposed DA algorithm exhibits high *flexibility*: by appropriately *tuning different control parameters*, the algorithm’s search behavior and overall performance can be *effectively shaped* (substantially *improved*).

— It would be valuable in the future work to experiment with *other types of improvement and/or perturbation algorithms* serving as the “descent” and “ascent” components.

— Another promising research direction would be to *integrate* the DA algorithm within *population-based or evolutionary frameworks*, such as genetic algorithms, memetic algorithms, or swarm intelligence methods, to *exploit synergistic hybridization effects*.

## REFERENCES

Burkard, R.E.; Karisch, S.; Rendl, F.: QAPLIB – a quadratic assignment problem library. J. Glob. Optim. 1997, 10, 391-403. http://www.seas.upenn.edu/qaplib (cited 10 September 2025

De Carvalho, Jr., S.A.; Rahmann, S. Microarray Layout as a Quadratic Assignment Problem. In German Conference on Bioinformatics, GCB 2006, Lecture Notes in Informatics – Proceedings, Vol.P-83; Huson, D.; Kohlbacher, O.; Lupas, A.; Nieselt, K.; Zell, A., Eds.; Bonn: Gesellschaft für Informatik, 2006, pp. 11-20.

Drezner, Z.; Hahn, P.M.; Taillard, E.D. Recent advances for the quadratic assignment problem with special emphasis on instances that are difficult for metaheuristic methods. Ann. Oper. Res., 2005, 139, 65-94. https://doi.org/10.1007/s10479-005-3444-z

Feo, T.A.; Resende, M.G.C. A probabilistic heuristic for a computationally difficult set covering problem. Oper. Res. Lett. 1989, 8, 67-71. https://doi.org/10.1016/0167-6377(89)90002-3

Misevičius, A.; Ostreika, A.; Verenė, D.; Andrejevas, A.; Žekienė, G. An improved hybrid genetic-hierarchical algorithm for the quadratic assignment problem. Mathematics 2024, 12, 3726. https://doi.org/10.3390/math12233726

Taillard – QAP. http://mistic.heig-vd.ch/taillard/problemes.dir/qap.dir/qap.html (cited 10 September 2025).