12th International Conference on Industrial Engineering

(ICIE 2016)

EngineeringIran
Institute of

Kharazmi University

12th International Conference on Industrial Engineering

A Greedy algorithm for a bi-team orienteering problem with time window

Amirhossein Karimi, Mahdi Bashiri

Industrial engineering department, Faculty of engineering, Shahed University, Tehran, Iran
Corresponding author: *E-mail address*: Bashiri.m@gmail.com.

Abstract

In this paper an algorithm is designed to solve the bi-team orienteering problem with time window (BTOPTW). Aim of this study is to minimize the total travel cost and maximize the score gained by the salesman. In the real cases, minimizing the salesmen total travel costs is very important for bosses. In this study, a greedy solution approach is proposed. Different approaches of extracting non-dominated solutions were analyzed. The algorithms were compared by some measures such as: non-dominated Pareto size (NPS), spacing metric SM), generational distance (GD) and inverse generational distance (IGD). The effect of the time window is discussed in this paper and a comparison for orienteering problem with and without time windows is discussed. The results confirm that the proposed greedy algorithm has a satisfactory efficiency.

Keywords: Team orienteering problem, Time window, Multi-objective, Greedy algorithm.

1-Introduction

The bi-orienteering problem was presented by Schilde [1] for the first time. The bi-orienteering problem is an extension of selective travelling salesman which was introduced by Tsiligirides [2]. In the orienteering problem each node has a visiting score. The aim is to find the tour which has the maximum score according to the time budget. For example a tourist has a time limitation for staying in a city and he/she wants to visit the best places in the time that he/she stays in that city. In the bi-orienteering problem presented by Schilde [1], there are two kinds of places in the city with different kinds of profits and the tourists want to maximize the score of both kinds of profits. Marketing of companies can be as an application of the orienteering problem. Marketing manager of a company wants to visit more important customers (with more profits) and minimize the total visiting costs too. In the real cases there is a time window for each customer and it is not possible to visit the customer according to the visitor preferred time schedule. In bigger companies, marketing will be done by more than one person. We start the paper with a literature review of the orienteering problem in second part. In section3, the mathematical model of the BTOPTW is presented and the augmented epsilon constraint and epsilon constraint are analysed for the BTOPTW. In section 4, the greedy algorithm is presented to solve the problem. In section 5, the ability of the algorithm is analysed and the instances of the BTOPTW are solved. Finally in the last section, the paper will be concluded.

2-Literature review

The name of orienteering problem originates from name of a game [3]. In the orienteering problem each node have special profit and the salesman want to gain the most score that he/she can gain in the time budget. One of the applications of the orienteering problem is a tourist who wants to visit historical places in a city and he/she cannot visit all of the historical places in the city because of the limitation of the time. The team orienteering problem is an extension of orienteering problem that there is more than one salesman, this extension of the orienteering problem was presented by Chao [4]. One of the applications of the team orienteering problem is needed for the section of marketing in a company. In this section there are many marketers that want to visit the richer customer with the priority and each of the marketers has time limitation for visiting the customer. The bi orienteering problem was presented by Schilde [1], in this problem there is two kinds of nodes that the salesman wants to gain the maximum benefits in both kind of nodes scores. A greedy procedure (GRASP) was designed for bi orienteering problem by Marti [5]. In table1, a review of the orienteering problem researches from 2010 until now which are related to the paper contribution is presented. In figure1, the increasing of recent researches about orienteering problem is shown.

Table1-The review of the orienteering problem (2010-2015)

Article	Solution method			Constraints	The number of salesmen		Type of the problem	
	Metaheuristic	Heuristic	Exact		Single	Team	OP[a]	Bi-[b]
(Souffriau,	√			...		√	√	
(Poggi, 2010			√	...		√	√	
(Bouly, 2010	√			...		√	√	
(Muthuswamy, 2011)[9]	√			....		√	√	
(Labadie, 2011	√			Time window		√	√	
(Labadie, 2012	√			Time window		√	√	
(Lin, 2012	√			Time window		√	√	
(Liang, 2013		√			√		√	
(Kim, 2013		√		....		√	√	
({Dang, 2013			√	....		√	√	
(Lin, 2013	√			...		√	√	
(Luo, 2013		√		...		√	√	
(Cura, 2014	√			Time window		√	√	
(Hu, 2014)[19]		√		Time window		√	√	
(Duque, 2015	√			Time windows	√		√	
(Martí, 2015	√				√			√
(Gunawan,	√			Time windows	√		√	
(Zabielski,	√			...	√		√	

[a] Orienteering problem
[b] Bi-orienteering problem

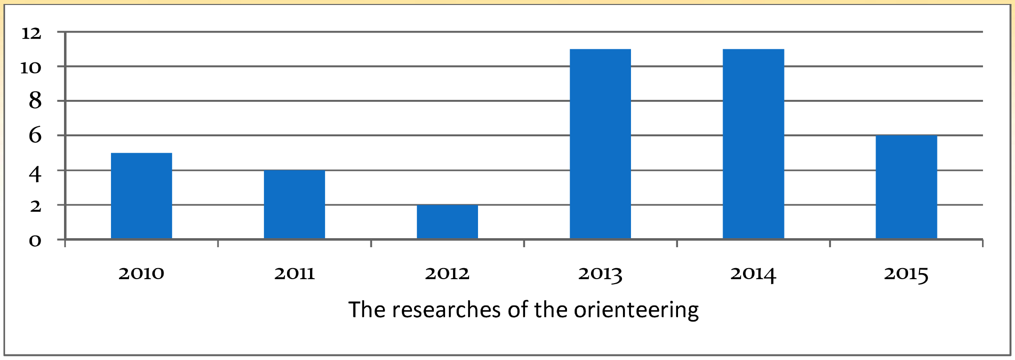

Figure1- Number of recent related researches about the orienteering problem

3- Mathematical model and the solution approach

In this section, the mathematical model of the proposed BTOPTW model is presented. In the BTOPTW, a time window $[O_i, C_i]$ is assigned to each node. The BTOPTW can be formulated in the following way: $X_{ijp} = 1$ if the person *p* travels from node *i* to *j*, and $y_{ip} = 1$ if the node *i* is visited by the person *p*, and s_{ip} is the starting time of the service in node *i* by person *p* and τ_i is the time of servicing in the node *i,* and *M* is a large constant value.

$$Max \sum_{i\in N}\sum_{p\in P} s_i y_{ip} \qquad (١)$$

$$Min \sum_{i=1}^{N-1}\sum_{j=2}^{N} t_{ij} x_{ijp} + \sum_{i\in N}\sum_{p\in P} \tau_i y_{ip} \qquad (2)$$

$$\sum_{p=1}^{P}\sum_{j=2}^{N} X_{1jp} = \sum_{p=1}^{P}\sum_{i=1}^{N-1} X_{iNp} = P \qquad (3)$$

$$\sum_{i=1}^{N-1} X_{ikp} = \sum_{j=2}^{N} X_{kjp} = y_{kp} \qquad \forall k = 2, \dots, N-1; \forall p = 1, \dots P \qquad (4)$$

$$S_{ip} + t_{ij} + \tau_i - s_{jp} \leq M\left(1 - x_{ijp}\right) \qquad \forall i,j = 1, \dots, N-1; \forall p = 1, \dots P \qquad (5)$$

$$\sum_{p=1}^{P} y_{kp} \leq 1 \qquad \forall k = 2, \dots, N-1 \qquad (6)$$

$$\sum_{i=1}^{N-1}\sum_{j=2}^{N} t_{ij} x_{ijp} + \sum_{i\in N}\sum_{p\in P} \tau_i y_{ip} \leq T_m \qquad \forall p = 2, \dots, p-1 \qquad (7)$$

$$O_i \leq S_{ip} \qquad \forall i = 1, \dots, N; \forall p = 1, \dots P \qquad (8)$$

$$S_{ip} \leq C_i \qquad \forall i = 1, \dots, N; \forall p = 1, \dots P \qquad (9)$$

$$X_{ijp}, y_{ip} \in \{0,1\} \qquad \forall i,j = 1, \dots, N; \forall p = 1, \dots P \qquad (10)$$

The objective function (1) maximizes total score. Function (2) minimizes the total cost which is assumed time dependent. Constraint (3) guarantees that all of the people start from node 1 and end in node *N.* Constraints (4) and (5) determine the connectivity and timeline of the tour of each person. Constraint (6) ensures that every vertex is visited at most once and Constraint (7) limits the time budget. Constraints (8) and (9) restrict the start of the service to the time window. The aim is to find the non-dominated solution of the problem, so we consider the epsilon constraint and an augmented epsilon constraint approaches to find the Pareto front.

3-1-Using of the epsilon constraint to find non-dominated solutions

The epsilon constraint is a method that finds the non dominated solutions for the problem. For finding the non dominated solutions, the best solution of each objective is needed. In another word, first we find the pay-off matrix.

$$MAX\, f_1 \quad (11)$$
$$MIN \quad f_2 \quad (12)$$
$$g(x)>0 \quad (13)$$

For each objective the problem should be solved. The epsilon constraint is solved in the formation which is in follow:

$$MAX\, f1 \quad (14)$$
$$f_2 \leq f_2^* + k\epsilon_i \quad (15)$$
$$g(x)>0 \quad (16)$$

ϵ_i is find by the follow equation:

$$\epsilon_i = \frac{m \quad (f_i) - m \quad (f_i)}{m \quad (k)} \quad (17)$$

$$Max\, f_1 \quad (18)$$

$$f_1 = \sum_{i \in N} \sum_{p \in P} s_i y_{ti} \quad (19)$$

$$f_2 = \sum_{i=1}^{N-1} \sum_{j=2}^{N} t_{i} x_{i} + \sum_{i \in N} \sum_{p \in P} \tau_i y_{ti} \quad (20)$$

$$f_2 \leq f_2^* + k\epsilon_i \quad (21)$$

Constraints (3) to (10)

3-2- Using of the augmented epsilon constraint to find non-dominated solutions

The augmented epsilon constraint is a reformed formation of the epsilon constraint. Previously other researchers used it to find non dominated solutions more efficiently [23]. By this method, it is guaranteed to find true Pareto front while in addition to the main objective other objectives are appeared in the main objective as well. The augmented epsilon constraint selects the best solution by affecting the other objectives in the main objective of model by a weight coefficient. The φ is a small constant between 10^{-3} and 10^{-6}.

$$MAX\, f_1 - \varphi(S_2 + S_3 + \cdots + S_K) \quad (22)$$
$$f_2 + s_2 = e_2 \quad (23)$$
$$g(x)>0 \quad (24)$$

e_i is find by the follow equation:

$$e_i = f_i^* - k\frac{m \quad (f_i) - m \quad (f_i)}{m \quad (k)} \quad (25)$$

$$Max\ f_1 - (\varphi S_1) \quad (26)$$

$$f_1 = \sum_{i \in N} \sum_{p \in P} s_i y_{ti} \quad (27)$$

$$f_2=\sum_{i=1}^{N-1}\sum_{j=2}^{N} t_{ij}\, x_{ij} + \sum_{i\in N}\sum_{p\in P} \tau_i y_{ip} \tag{28}$$

$$f_2 + S_2 = e_2 \tag{29}$$

Constraints (3) to (10)

4-Proposed greedy algorithm to solve the BTOPTW

In this part, the greedy algorithm is presented for optimizing of the bi- team orienteering problem with time window. In the greedy randomized adaptive search procedure (GRASP), the aim is to make a solution. But in the proposed greedy algorithm, appropriate solutions are selected from the initial solutions which were constructed randomly. Each node can be visited during its time window that is assigned to that node and there is a time budget for each person for visiting nodes. Each infeasible solution will be deleted and replaced by its feasible neighbour solution. The greedy algorithm is applied in two stages. First the aim is maximizing the score and the selected solutions are put in set *x* and after that the aim is minimizing the cost and the selected solutions are put in the set *y*. Both of the *x* and *y* sets are put in set *xx*. The cost and the score of each solution are calculated and the non-dominated solutions are selected. The Pseudocode of the greedy algorithm for optimizing the BTOPTW is presented in algorithm1.

ALGORITHM1: Greedy algorithm for the BTOPTW

Step1: Generate initial solution
Step2: $x \leftarrow \emptyset$
Step3: $C \leftarrow in$ sc
Step4: **while** greedy algorithm stopping criterion not satisfied do
Step5: for all c∈ C compute the *g(c)*=score*, gmax=max(g(c))* and *gmin=min(g(c))*
Step6: define RCL← $g(c) \geq g$ $- \alpha(gm$ $- g$) and α∈ [0,1]
Step7: select c^* randomly from RCL
Step8: add c to *x*
Step9: **endwhile**
Step10: **for** all solutions
Step11: **if** the solution is infeasible
Step12: apply a repair procedure
Step13: **endif**
Step14: **endfor**
Step15: $y \leftarrow \emptyset$
Step16: $C \leftarrow in$ sc
Step17: **while** greedy algorithm stopping criterion not satisfied do
Step18: for all c∈ C compute the g(c)=cost, *gmax=max(g(c))* and *gmin=min(g(c))*
Step19: define RCL← $g(c) \leq g$ $+ \alpha(g$ $- g$) and α∈ [0,1]
Step20: select c^* randomly from RCL
Step21: add c to *y*
Step22: **endwhile**
Step23: **for** all solution
Step24: **if** the solution is infeasible
Step25: apply a repair procedure
Step26: **endif**
Step27: **endfor**
Step28: *xx*← $[x, y]$
Step29: Extract the non dominated solutions from *xx*

Four ways are presented in this paper for finding neighbourhood solutions which are insertion, deletion, swap- in, swap- out explained as following.

In insertion, it is assumed that there are some nodes that can be added to the tour, one of them is selected randomly and will be placed randomly in the tour, but the tour of each person should be started from node 1 and end at node *N* as depicted in Figure2 (a). In deletion, one of the nodes that can be deleted from the tour (all nodes except 1, N) is selected at random and will be deleted from the tour as depicted in Figure2 (b). In swap-in, two nodes of the tour (except 1, N) are selected and will be swapped as depicted in Figure2 (c). In swap-out, one of the nodes of tour (except 1, N) is selected, and one of the nodes that can be added to the tour is selected too. These two nodes will be swapped as depicted in Figure2 (d).

1	3	5	1	5	
1	3	5	1	4	5

2	4
2	

(a)

1	3	5	1	5
1	5	1	5	

(b)

1	3	5	1	4	5
1	4	5	1	3	5

(c)

1	3	5	1	5
1	2	5	1	5

2	4
3	4

(d)

Figure2_Neighborhood solution generation schemes. (a) Insertion (b) Deletion (c) Swap-in (d) swap-out

5- Computational experiments

In this section, the performance of the proposed algorithm is considered. To consider the performance of the algorithm, the results are compared with the results of an exact solution. To check the efficiency of the algorithm, instances were solved by the proposed algorithm and by the CPLEX solver. The results have been reported in table4. The code was written in MATLAB 2014 and all computations have been run on a PC with following specifications: Intel(R) Core(TM) i3 CPU processor 2.53 gigahertz, 2 gigabytes RAM. In this section we analysed the performance of the epsilon constraint and it augmented version for different instances. Then we study the performance of our proposed algorithm with determined exact Pareto front.

5-1-Comparison of the augmented epsilon constraint and epsilon constraint

In this section, the ability of epsilon constraint and the augmented epsilon constraint is compared. Two measures are used for comparing the augmented epsilon constraint and epsilon constraint in this paper such as non-dominated solutions NPS and SM and their computation equations are presented in (30) and (31). The algorithm with the most value of NPS is the better one while the preferred algorithm has smaller SM measure. There are *n* solutions and f_k^i is the response of the function *k* for solution *i*. d^- is the average of the *d*.

$$SM=\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(d^- - d_i)^2} \tag{30}$$

$$d_i = m_{\;j\in N \;\wedge j\neq i} \sum_{k=1}^{k} |f_k^i - f_k^j| \tag{31}$$

Eight instances were solved by epsilon constraint and augmented epsilon constraint and the results of solving these instances are shown in table 2. The results of table 2 are compared by mentioned measures in table 3. As it was shown in table3, augmented epsilon constraint is better approach for finding the non-dominated solutions than the epsilon constraint.

Table2-The comparison of extracted non dominated solutions by two approaches

Instance	Epsilon constraint		Augmented epsilon constraint	
	f1	*f2*	*f1*	*f2*
p1-05	0	104	36	323
	12	139	28	314
	20	178	20	178
	28	314	12	139
	36	323	0	104
pr1_07	0	146	53	423
	16	175	48	395
	24	222	40	388
	36	342	36	342
	41	349	28	303
	48	395	24	224
	53	423	16	175
	-	-	0	146
pr1_08	0	36	58	487
	8	113	50	440
	12	160	46	393
	21	196	36	342
	29	244	29	244
	36	342	21	196
	46	393	12	160
	50	440	8	113
	58	487	0	36
pr1_09	0	224	71	468
	24	231	66	440
	37	268	53	411
	42	293	50	331
	45	305	45	305
	50	331	42	293
	53	411	37	268
	66	440	24	231
	71	468	0	224
p1_10	0	56	46	361
	13	85	42	296
	21	133	34	235
	34	235	21	133
	42	296	13	85
	46	361	0	56
pr1_11	0	84	33	389
	13	117	25	328
	21	177	21	177
	25	328	13	117
	33	389	0	84
pr1_12	0	100	58	417
	33	124	53	384
	41	183	46	246
	46	246	41	183
	53	384	33	124
	58	417	0	100
pr1_13	0	158	13	177
	13	177	0	158

Table3-The performance comparison of two approaches by mentioned measures

Instance	Epsilon constraint		Augmented epsilon	
	SM	NPS	SM	NPS
pr1_05	0.0273	5	0.0273	5

pr1_07	0.0225	7	0.0154	8
pr1_08	0.0112	9	0.0112	9
pr1_09	0.0172	9	0.0172	9
pr1_10	0.0111	6	0.0111	6
pr1_11	0.0131	5	0.0131	5
pr1_12	0.0429	6	0.0429	6
pr1_13	0	2	0	2

5-2- Analysis of the proposed algorithm performance in determining the Pareto front

In this section the ability of the proposed algorithm will be discussed. First the measure that is used for comparing the proposed algorithm is presented and then the results of numerical instances are reported.

Two measures are used in this paper that can show the ability of the algorithm which is proposed in this study. These two measures are Generational Distance (GD) and Inverse Generational Distance (IGD).

GD calculates the distance of the non-dominated solutions of the algorithm from the non-dominated solutions of an exact approach. GD is based on the convergence of the solutions from a real Pareto front and its measure is presented in equation (32). The small value of GD shows the accuracy of the algorithm.

Y' is the set of the extracted non-dominated solutions by the proposed algorithm. $\overline{Y}$ is the set of the real non-dominated solutions extracted by the augmented epsilon constraint. L is the number of objectives.

$$GD(Y') = \frac{1}{|Y'|}\sum_{P'\in Y'}(\min_{\overline{q}\in\overline{Y}}\sqrt{\sum_{i=1}^{L}(f_i(p') - f_i(\overline{q}))^2}) \tag{32}$$

The algorithm which has less *GD* is better algorithm and when *GD*=0, it means that the algorithm has found all solutions of the real Pareto front. In contrast, *IGD* is based on the diversity of the extracted solutions and its measure is presented in equation (33). The small value of the IGD shows the accuracy of the algorithm.

$$IGD(Y') = \frac{1}{|\overline{Y}|}\sum_{\overline{q}\in\overline{Y}}(\min_{p'\in Y'}\sqrt{\sum_{i=1}^{L}(f_i(\overline{q}) - f_i(p'))^2}) \tag{33}$$

To show the performance of the proposed algorithm, all of its extracted non-dominated solutions were compared by the exact Pareto front using the GD, IGD, NPS and SM metrics. The results have been shown in table 5.

5-3- The effect of the time window in the team orienteering problem

The time windows constraint makes the salesman to visit the customers in the specific time scope in a day which makes the travelling hard. As it is shown in the table5, the ordinary team orienteering problem has more scores than the Team orienteering problem with time windows while its results contain less customer satisfaction because of violation of their preferred time window. This comparison confirms that the proposed model is a valid one.

Table4-The measures for exact algorithm and greedy algorithm

Instance	Epsilon constraint		Greedy algorithm			
	SM	NPS	SM	NPS	GD	IGD
pr1_05	0.0225	7	0.0273	5	0	0
pr1_07	0.0225	7	0.0124	8	15.625	30.7143
pr1_08	0.0112	9	0.0114	11	14.1818	24

pr1_09	0.0172	9	0.0105	10	24.8	39.3333
pr1_11	0.0131	5	0.0333	3	1.8	180
pr1_12	0.0697	5	0.0223	7	14.4286	72.6
pr1_13	0	2	0	2	0	0

Table5-The effect of the time window

Instance	TOPTW		TOP	
	f1	*f2*	*f1*	*f2*
pr1_10	46	361	91	803
pr1_11	33	389	104	1044
pr1_12	58	417	124	1013
pr1_13	13	177	137	879

6-Conclusion

In this paper, a modified bi team orienteering problem with time windows was proposed considering maximizing the score and minimizing the cost which is dependent to the time of the tour. Then the ability of the epsilon constraint and augmented epsilon constraint was studied and compared by some measures. The experiments on some instances confirm that the augmented epsilon constraint never find the worse solutions than the epsilon constraint. Also the ability of the greedy algorithm is analyzed by some measures as well. The results confirm the efficiency of the proposed greedy solution approach. Finally some sensitivity analysis especially the time window confirms the model validity. As a future study, using of the greedy algorithm hybridized by other reliable solution approaches such as NSGA-II is suggested to solve the problem in larger instances.

References

[1] Schilde, M., K. F. Doerner, et al. (2009). "Metaheuristics for the bi-objective orienteering problem." Swarm Intelligence **3**(3): 179-201.
[2] Tsiligirides, T. (1984). "Heuristic methods applied to orienteering." Journal of the Operational Research Society: 797-809.
[3] Chao, I. (1993). "Algorithms and solutions to multi-level vehicle routing problems."
[4] Chao, I.-M., B. L. Golden, et al. (1996). "The team orienteering problem." European Journal of Operational Research **88**(3): 464-474.
[5] Martí, R., V. Campos, et al. (2015). "Multiobjective GRASP with Path Relinking." European Journal of Operational Research **240**(1): 54-71.
[6] Souffriau, W., P. Vansteenwegen, et al. (2010). "A path relinking approach for the team orienteering problem." Computers & Operations Research **37**(11): 1853-1859.
[7] Poggi, M., H. Viana, et al. (2010). "The team orienteering problem: Formulations and branch-cut and price." 10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems: 142.
[8] Bouly, H., D.-C. Dang, et al. (2010). "A memetic algorithm for the team orienteering problem." 4OR **8**(1): 49-70.
[9] Muthuswamy, S. and S. S. Lam (2011). "Discrete particle swarm optimization for the team orienteering problem." Memetic Computing **3**(4): 287-303.
[10] Labadie, N., J. Melechovský, et al. (2011). "Hybridized evolutionary local search algorithm for the team orienteering problem with time windows." Journal of Heuristics **17**(6): 729-753.
[11] Labadie, N., R. Mansini, et al. (2012). "The team orienteering problem with time windows: An lp-based granular variable neighborhood search." European Journal of Operational Research **220**(1): 15-27.
[12] Lin, S.-W. and F. Y. Vincent (2012). "A simulated annealing heuristic for the team orienteering problem with time windows." European Journal of Operational Research **217**(1): 94-107.
[13] Liang, Y.-C., S. Kulturel-Konak, et al. (2013). "A multiple-level variable neighborhood search approach to the orienteering problem." Journal of Industrial and Production Engineering **30**(4): 238-247.
[14] Kim, B.-I., H. Li, et al. (2013). "An augmented large neighborhood search method for solving the team orienteering problem." Expert Systems with Applications **40**(8): 3065-3072.
[15] Dang, D.-C., R. N. Guibadj, et al. (2013). "An effective PSO-inspired algorithm for the team orienteering problem." European Journal of Operational Research **229**(2): 332-344. [16] Lin, S.-W. (2013). "Solving the team orienteering problem using effective multi-start simulated annealing." Applied Soft Computing **13**(2): 1064-1073.

[17] Luo, Z., B. Cheang, et al. (2013). "An adaptive ejection pool with toggle-rule diversification approach for the capacitated team orienteering problem." European Journal of Operational Research **229**(3): 673-682.
[18] Cura, T. (2014). "An artificial bee colony algorithm approach for the team orienteering problem with time windows." Computers & Industrial Engineering **74**: 270-290.
[19] Hu, Q. and A. Lim (2014). "An iterative three-component heuristic for the team orienteering problem with time windows." European Journal of Operational Research **232**(2): 276-286.
[20] Duque, D., L. Lozano, et al. (2015). "Solving the Orienteering Problem with Time Windows via the Pulse Framework." Computers & Operations Research **54**: 168-176.
[21] Gunawan, A., H. C. Lau, et al. (2015). An Iterated Local Search Algorithm for Solving the Orienteering Problem with Time Windows. Evolutionary Computation in Combinatorial Optimization, Springer**:** 61-73.
[22] Zabielski, P., J. Karbowska-Chilinska, et al. (2015). A Genetic Algorithm with Grouping Selection and Searching Operators for the Orienteering Problem. Intelligent Information and Database Systems, Springer**:** 31-40.
[23] Du, Y., L. Xie, et al. (2014). "Multi-objective optimization of reverse osmosis networks by lexicographic optimization and augmented epsilon constraint method." Desalination **333**(1): 66-81.