Polygon regular location problem

Lýdia Gábrišová[1], Jaroslav Janáček[2]

Abstract. The polygon regular location problem originates in public transportation system management, where a coordination of circulating vehicles must be performed so that the vehicle arrivals are distributed regularly in some period. The contribution deals with the problem complexity and reports on various ways of reducing the computational effort, which is necessary to obtain an optimal solution of the problem using a general optimization environment.

Keywords: regular polygon, location problem, public transport, coordination of vehicle arrivals, free order of objects.

JEL Classification: C61
AMS Classification: 90C27

1 Introduction

The problem originated in the field of public transport as we can see in [2, 4, 8]. An original goal was to increase attractiveness of public transport by making schedule of urban and sub-urban transport more regular at some selected stops. It was taken into account that a regularity of vehicle arrivals optimizes a transportation supply for passengers by non-investment way, which can be seen in [1, 3, 5, 8]. It was found that individual vehicles as buses, trams or trolleybuses circle along their lines in the associated urban transportation network and in addition an average time of traversing a cycle is relatively short. Under these circumstances, the same vehicle usually appears at an observed stop several times in a given period. All the vehicle arrivals form a transportation supply for the passengers coming at the stop. If some arrivals follow closely one after other, the second one of the arrivals does not contribute considerably to the transportation supply.

On the other hand, long intervals between arrivals cause an unpleasant time loss for passengers, which come at the stop randomly. It follows that some regularity of the arrivals is desirable. The regularity of vehicle arrivals at the given stop can be improved by a shift of arrival time of an individual vehicle. As a given vehicle appears at the stop several times in the given period, a shift of one of its arrivals causes shifts of all its arrivals in the period by the same value. Furthermore, it must be considered that the observed stop can be served by vehicles, which traverse different lines. It causes that time intervals between neighboring arrivals of different vehicles differ. If the observed period is long enough to be divisible by circle time of each considered vehicle, then arrival times of a given vehicle can be depicted as vertices of a regular polygon on a circle, whose circumference is equal to the length of the period. That is for; we can call the problem as location of vertices of polygons on a circle or briefly the regular polygon location problem. In the problem, the goal is to locate the set of regular polygons in a circle so that all vertices lie on the same circumference and their distribution be regular [6, 7, 11].

Many researchers tackled this problem in several recent decades but due to non-linearity and discreteness of associated models, only heuristics have been used to solve this problem. In this contribution, we present an exact approach based on usage of particular characteristics of the problem and thorough model building. This approach together with new possibilities offered by used optimization environment enable us to solve some instances of the problem to optimality, which can be seen in [9, 10, 13].

2 Reduced Formulation of the Regular Polygon Location Problem

Let us consider r regular polygons with the same radius and center. It follows that all polygon vertices lie on the circumference. Let the p-th polygon have n_p vertices. Vertex locations of the polygon p on the circle are uniquely given by an angle between a zero point on the circle and the first vertex of the polygon. Let T denote the circumference of the circle given in some angle units and let $d_p=T/n_p$ hold. If we introduce a decision variable x_p, which denotes the angle between the zero point and the first vertex of the p-th polygon, then the second vertex has location d_p+x_p, the third vertex has location $2d_p+x_p$ and so on.

[1] University of Žilina, Faculty of Management Science and Informatics, Department of Mathematical Methods, Univerzitná 2, 010 26 Žilina, Slovak Republic, email: lydia.gabrisova@fri.uniza.sk
[2] University of Žilina, Faculty of Management Science and Informatics, Department of Transportation Networks, Univerzitná 2, 010 26 Žilina, Slovak Republic, email: jaroslav.janacek@fri.uniza.sk.

In general the variable x_p corresponds with a value of rotation of the p-th polygon from a zero point. We also define a reverse mapping $p(k)$, which returns the index of polygon containing the vertex k (see figure 1). The total number of involved vertices is denoted as m. It is obvious that range $< 1, d_p >$ is sufficient for x_p, $p \geq 2$, to cover all possible locations of the p-th polygon vertices taking into consideration the condition that no vertex location is allowed to meet other vertex locations.

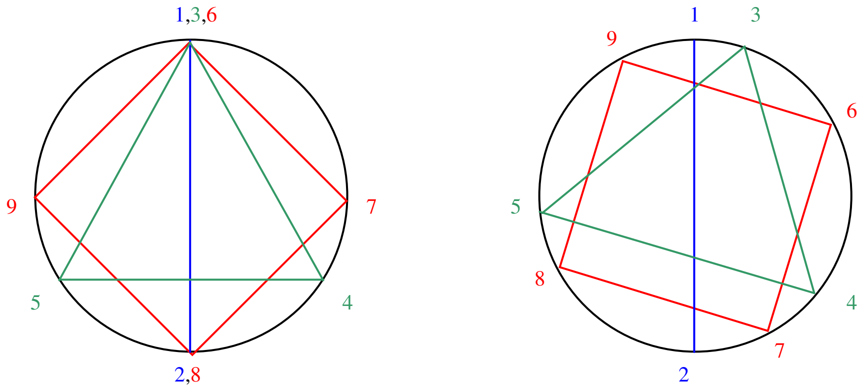

Figure 1 Example of three polygons $n_1 = 2$, $n_2 = 3$, $n_3 = 4$ vertices with designed labelling of vertices. Left figure – the default state and right figure – possible rotation of polygons, where the first polygon is fixed.

Now we can assign the lowest value of rotation a_k to each vertex $k = 1, \ldots, m$ and state that current value of rotation of vertex k is given by $a_k + x_{p(k)}$ and this location varies over range $< a_k, a_k + d_{p(k)} - 1)$. Without loss of generality, we can set the values of a_1 and x_1 at zero. The regular polygon location problem can be formulated as a search for such vector $< 0, x_2, \ldots, x_r >$, which corresponds to the most regular distribution of the vertices along the period T (circumference). The regularity is considered as a sum of squares of the differences between neighboring vertex locations in this contribution. If we denote t_k as circumference distance between the vertex k and the directly preceding vertex, then sum of $(t_k)^2$ for $k = 1, \ldots, m$ corresponds with convex nonlinear objective function. The k-th item of the objective function can be linearized in accordance to [12].

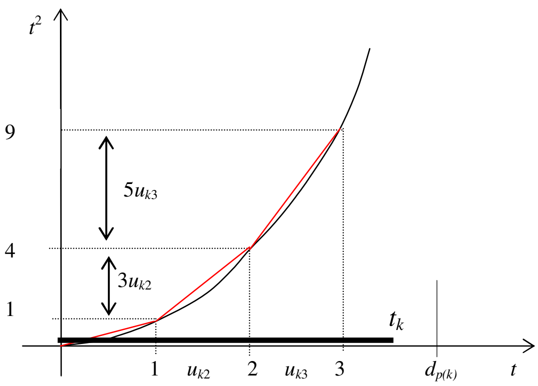

Figure 2 Piecewise linearization of the item $(t_k)^2$.

For t_k, $k = 1, \ldots, m$ we introduce non-negative auxiliary variables $u_{kj} \leq 1$ for $j = 2, \ldots, d_{p(k)}$. Then we can express t_k as the sum of u_{kj} in accordance to (1) as depicted in figure 2. The square of t_k can be replaced by linear expression depending on variables u_{kj} accordingly to (2).

$$t_k = 1 + \sum_{j=2}^{d_{p(k)}} u_{kj} \quad \text{for } k = 1,..,m \tag{1}$$

$$(t_k)^2 = 1 + \sum_{j=2}^{d_{p(k)}} (2j - 1) u_{kj} \qquad \text{for } k = 1,..,m \tag{2}$$

If the values of u_{kj} are integer, then the right-hand-side of (2) represents the exact value of $(t_k)^2$. Let index of the vertex preceding the vertex k be denoted as $i(k)$. We realize that the variable x_1 was set to zero. This way, t_1 can be defined as $T - a_{i(1)} - x_{p(i(1))}$, where $i(1)$ is index of the vertex with the biggest value of location in the period T. The other variables t_k must satisfy the link-up constraints $t_k = a_k + x_{p(k)} - a_{i(k)} - x_{p(i(k))}$ for $k = 2, \ldots, m$. Unfortunately the order of vertices changes by jumps, when the values of x_p vary.

The non-linearity involved in precedence mapping (permutation) *i(k)* used in the substitution constraints for t_k can be removed by introducing auxiliary zero-one variables $w_{ik} \in \{0, 1\}$ for each relevant pair (i,k), $i = 1, \ldots, m$, $k = 1, \ldots, m$, $i \neq k$. A variable w_{ik} takes the value of one if and only if the vertex i directly precedes the vertex k. To describe the relevant pairs in the following model, we introduce a logical function *exists* defined on all pairs $(i,k) \in \{1, \ldots, m\} \times \{1, \ldots, m\}$. The function *exists(i,k)* takes the value of true, if and only if the pair (i,k) is relevant. Making use of the previously introduced variables x_p and u_{kj} a reduced linear model of the regular polygon location problem can be formulated as follows:

$$\textit{Minimize} \quad \sum_{k=1}^{m} (1 + \sum_{j=2}^{d_{p(k)}} (2j-1) u_{kj}) \tag{3}$$

$$\textit{Subject to} \quad \sum_{\substack{i=1 \\ exists(i,k)}}^{m} w_{ik} = 1 \quad \textit{for } k=1,..,m \tag{4}$$

$$\sum_{\substack{i=1 \\ exists(k,i)}}^{m} w_{ki} = 1 \quad \textit{for } k=1,..,m \tag{5}$$

$$x_1 = 0,\ x_p \le d_p - 1 \qquad \textit{for } p=2,..,r \tag{6}$$

$$T - a_i - x_{p(i)} \ge 1 + \sum_{j=2}^{d_{p(1)}} u_{1j} - T^L_{i1} * (1 - w_{i1}) \quad \textit{for } i=2,..,m,\ exists(i,1) \tag{7}$$

$$T - a_i - x_{p(i)} \le 1 + \sum_{j=2}^{d_{p(1)}} u_{1j} + T^U_{i1} * (1 - w_{i1}) \quad \textit{for } i=2,..,m,\ exists(i,1) \tag{8}$$

$$a_k + x_{p(k)} - a_i - x_{p(i)} \ge 1 + \sum_{j=2}^{d_{p(k)}} u_{kj} + T^L_{ik} * (1 - w_{ik}) \ \textit{for } k=2,..,m,\ i=1,..,m,\ exists(i,k) \tag{9}$$

$$a_k + x_{p(k)} - a_i - x_{p(i)} \le 1 + \sum_{j=2}^{d_{p(k)}} u_{kj} + T^U_{ik} * (1 - w_{ik}) \ \textit{for } k=2,..,m,\ i=1,..,m,\ exists(i,k) \tag{10}$$

$$u_{kj} \le 1,\ u_{kj} \ge 0 \qquad \textit{for } k=1, \ldots, m,\ j=2,..,d_{p(k)} \tag{11}$$

$$x_p \in Z^+ \qquad \textit{for } p=1,..,r \tag{12}$$

$$w_{ik} \in \{0,1\} \quad \textit{for } i=1,..,m,\ k=1,..,m,\ exists(i,k) \tag{13}$$

The consistency constraints (4) and (5) ensure that each vertex k has its predecessor and successor. The constraints (7) – (10) cause that if $w_{ik} = 1$ for some pair (i, k), then the difference between the location of vertex k and the location of preceding vertex i is equal to t_k given by substituting equality (1). If $w_{ik}=0$ holds, then the associated constraints are relaxed by suitable values of T^L_{ik} and T^U_{ik} .

3 Model Adjustments

In the model above, there are several loose ends, which must be set up before submitting an associated instance to a general IP-solver for solving. The first thing, which must be determined is the logical function *exists(i,k)*. The cardinality of the set of pairs *(i,k)*, for which the function takes the value of "*true*" corresponds with the number of binary variables w_{ik} and this number forms a considerable part of the instance size. As a general IP-solver performs common branch-and-bound method, it is obvious that complexity of the problem puts a tight limit on size of solved problem instances. The set of variables can be defined in several ways.

The basic approach introduces a variable w_{ik} for each pair *(i,k)*, where $i \neq k$. The approach with reduced model introduces a variable w_{ik} for each pair *(i,k)*, where inequality $a_i \le a_k + d_{p(k)} - 1$ holds. The advanced approach defines a variable w_{ik} for each pair *(i,k)*, where the constraints from the reduced approach is satisfied and, in addition, there exists no vertex j for which the following inequalities $a_i + d_{p(i)} \le a_j$ and $a_j + d_{p(j)} \le a_k$ hold.

The second loose end represents the lower and upper bounds of T^L_{ik} and T^U_{ik} respectively. These coefficients may influence the starting lower bound in the branch-and-bound computational process as the associated lower bounding uses *LP*-relaxation of the model.

The third loose end is the input order (numbering) of the polygons, especially the polygon, whose x_p can be fixed to zero. We can order the polygons either in descending or ascending order accordingly to their number n_p of vertices. In the first case, the polygon with the biggest number of vertices is fixed and in the second case the polygon with the smallest number can be fixed.

At the end of this section, we note that if we do not insist on the condition that no vertex location can share locations of the other vertices, then the constraints (7) and (9) can be relaxed. The original model will be called the full model and the reduced one will be referred as the half model. In the next section we try to show, how these possible settings may influence computational time of the used IP-solver.

4 Numerical Experiments

The presented numerical experiments are aimed at inspecting a special phenomenon, which occurred, when preliminary experiments [12] were performed. It was found that a reduction of decision variables did not impact the computational time proportionally. The numerical experiments were performed with two pools of instances, where each of the pools contains exactly six items. Each instance consists of one quadruple of polygons defined on a circle with circumference T=360. The total number of vertices included in one instance varies from 14 to 20 and from 21 to 27 for the first and second pools respectively. All experiments were performed using the optimization software FICO Xpress 7.1 (64-bit, release 2010). The associated code was run on a PC equipped with the Intel Core i5 2430M processor with the parameters: 2.4 GHz and 4 GB RAM.

The first series of experiments was performed with the full model. The values of T^L_{ik} and T^U_{ik} were set at the biggest value and descending order of polygons was used. The tested models differ only in the number of w_{ik}, where three cases “basic”, “reduced” and “advanced” are distinguished in accordance to the denotation introduced in the previous section. Each problem instance was solved to optimality and the associated average computation times and average numbers of introduced variables w_{ik} are reported in table 1.

		Basic		**Reduced**		**Advanced**	
Pool	**Range**	***Avg_CT***	***Avg_w*** Avg_w_{ij}	***Avg_CT***	Avg_w_{ij}	***Avg_CT***	Avg_w_{ij}
1	14-20	7.7	282	10.4	171	13.2	144
2	21-27	79.6	571	14.6	330	26.5	217

Table 1 Average computational times in seconds and average numbers of introduced variables are reported in columns denoted as Avg_w_{ij} and *Avg_CT* respectively.

The next portion of experiments was focused on the influence of the less or more tight adjustment of the bounds T^L_{ik} and T^U_{ik} on the computational time. The associated experiments were performed with the full model; the number of variables w_{ik} was reduced in accordance to the approach “advanced” and where descending order of polygons was used. Accordingly to the definition in the previous section, the values of the T^L_{ik} and T^U_{ik} were subsequently set at the compromise setting. Each instance of the pools was solved to optimality and the associated average computational times are plotted in table 2.

			Biggest s.	**Tight s.**	**Compromise s.**
Pool	**Range**	Avg_w_{ij}	***Avg_CT***	***Avg_CT***	***Avg_CT***
1	14-20	144	13.2	9.9	2.0
2	21-27	217	26.5	27.0	3.3

Table 2 Average computational times in seconds and average numbers of introduced variables are reported in columns denoted as Avg_w_{ij} and *Avg_CT* respectively.

The last portion of experiments concerns an impact of the possible reduction of constraint set and the polygon order to the performance of branch-and-bound search. In these experiments the biggest setting of T^L_{ik} and T^U_{ik} was applied and the approach “advanced” to introduction of variables w_{ik} was used. The ascending order was tested on the full model and the half model was combined with descending ordering of polygons. The resulting average computational times are reported in table 3.

Model			Full	Full	Half
Ordering			Descending	Ascending	Descending
Pool	**Range**	***Avg_w**$_{ij}$*	***Avg_CT***	***Avg_CT***	***Avg_CT***
1	14-20	144	13.2	3.4	374.2
2	21-27	217	26.5	7.5	2261.4 *)

Table 3 Average computational times in seconds and average numbers of introduced variables are reported in columns denoted as Avg_w_{ij} and *Avg_CT* respectively. *) The average was computed from five instances only. Computation of the sixth instance exceeded one hour and was prematurely terminated.

5 Conclusions

To our great surprise the obtained numerical results show that neither variable reduction nor constraint relaxation in model of the polygon regular location problem accelerate common branch-and-bound search embedded in an optimization environment. The only exception is the reduction from “basic” to “reduced” for the pool with bigger number of vertices. The similar results appear in the attempts for tighter setting of relaxing constants T^L_{ik} and T^U_{ik}. Nevertheless, it was found that a convenient setting, which compromises the loose and tight settings, might considerably reduce the computational time of the searching process. Furthermore, the ascending order of polygons in the input data also influences a computational time reduction.

Acknowledgements

This work was supported by the research grant of the Faculty of Management Science and Informatics, University of Žilina VEGA 1/0296/12 “Public Service Systems with Fair Access to Service” and APVV-0760-11 "Designing of Fair Service Systems on Transportation Networks". We would like to thank to "Centre of excellence in computer sciences and knowledge systems (ITMS 26220120007) for built up infrastructure, which was used.

References

[1] Brucker, P. and Hurink, J.: A railway-scheduling problem. In: *Zeitschr. f. Oper. Res.*30 (1986), A 223-A 227.

[2] Burkard, R. E.: Optimal schedules for periodically recusrring events. In: *Discrete Applied Mathematics* 15 (1986), 167-180.

[3] Černá, A.: Optimization of periodic and transport offered. *Scientific Papers of the University of Pardubice, series* B 4 (1998), 193-194.

[4] Černý, J.: Optimization of Periodic Time Tables in Railway and Urban Transport. In: *Proceedings of “Optimization in Production and Transportation”, Scheveningen* (1994)

[5] Černý, J. and Blaško, R.: A Note to Coordination of Periodic Processes in Transport. In: *TRANSCOM* 95, *University of Žilina* (1995), 95-98.

[6] Černý, J. and Guldan, F.: Location of polygon vertices on circles and its application in transport studies. *Apl. mat.* 32 (1987), 81-95.

[7] Černý, J. and Kluvánek, P.: *Basises of mathematical transport theory.* Veda, Bratislava, 1991 (in Slovak).

[8] Černý, J. and Mojžíš, V.: Note on the coordination of periodic processes in transportation systems. *Communications–Scientific Letters of the University of Žilina* 7/2 (2005), 25-27.

[9] Gábrišová, L. and Janáček, J.: The solution of some problems with free order of objects with IP-Solver. In: *Proceedings of workshop “Problems of Discrete Optimization for Transport”, Univerzity of Pardubice* (2012), (to apear).

[10] Gábrišová, L. and Kozel, P.: Accuracy of linear approximation used in non-linear problem optimization. *Journal of Information, Control and Management Systems of the University of Žilina Vol* 8, No 4 (2010), 301-309.

[11] Guldan, F.: Maximization of distances of regular polygons on a circle. *Apl.Mat.*25 (1980), 182-195.

[12] Janáček, J. and Gábrišová, L.: Regular polygon location problem with IP- Solver. In: *Proceedings of conference Quantitative methods in economic, University of Economics, Bratislava* (2012), 103-107.

[13] Kozel, P. and Gábrišová, L.: Minimalization of passengers waiting time by vehicle schedule adjustment. In: *TRANSCOM* 2011*, University of Žilina* (2011), 103-106.