

**First results of the hybrid object-oriented system
for the optimization of the construction site layout: ESBE**

Kunibert Lennerts

Institute of Construction Management and Machinery, University of Karlsruhe, Am Fasanengarten, 76128 Karlsruhe, Germany

ABSTRACT

ESBE is a hybrid object-oriented system. The objective is the interactive generation of an optimized construction site layout. The task is the locating of the individual construction site facilities. ESBE consists of three major parts: an object-oriented construction site database which contains general and company specific data of the construction site facilities and the building data; a hybrid system with mathematical optimization algorithms and a knowledge based system as well as a well-tailored user-interface with a 3D visualization component.

The output of the knowledge-based system is a ranking of the suitable areas (rectangles) in order to place the facilities. These areas serve as input to the mathematical algorithms, computing the optimal locations of the facilities inside these restricted areas. Both parts have been programmed with the object oriented programming language Eiffel. The first results of the developed heuristics are very promising.

1. INTRODUCTION

The aim of ESBE is the generation of an „optimal“ site layout considering the very complex and dynamic construction process. Thereby it would be possible to quantify site layouts and to evaluate different ones. This paper describes an heuristics procedure to find the „optimal“ locations of cranes throughout the different construction stages. The problem of crane location has been considered, because in Germany almost exclusively cranes are used for the transportation of materials. Rodriguez-Ramos¹ and Warszawski^{2,3} already dealt with this problem. However, Rodriguez-Ramos only determined a zero position of the crane. This is not a crane location technique in the original meaning of the word. Besides, he did not consider multiple phases. On a general basis, Warszawski considers multiple phases, but he only deals with the crane location problem within the framework of a single-phase project with only one facility and one kind of material. Both authors do not consider the feasibility of the reached solution.

An heuristics procedure had to be developed because exact algorithms in the area of the quadratic assignment problem have an NP-hard complexity.

2. MODEL OF THE HEURISTICS FOR THE DETERMINING OF A FEASIBLE CRANE LAYOUT WITH MINIMAL COST

Considering the following constraints

1. *Spatial constraints*: Within each phase of the construction process it has to be considered that all of the execution elements (EE) which have to be served must completely overlap with the jib range of the cranes momentarily located on the site (**overlap problem**)⁴. ESBE divides the entire building into execution elements (EE). These are cuboids which are assigned geometric, temporal (periods of execution) and mass-related (e.g. formwork, reinforcement, concrete) data. Furthermore, one has to make sure that the serving crane is capable of transporting the necessary masses to the execution elements (**weight problem**).
2. *Temporal constraints*: For all stages of construction it has to be valid that the transportation efforts having to be carried out by a crane do not exceed the time provided for this crane while taking into account the residence time. For example, within a construction stage taking as long as a working day it is only permissible to assign tasks to the crane which, together, do not need more than $8 \cdot 3600s = 28800s$ (**execution problem**).

the possible crane layout has to be determined with minimal cost. Provided that the dimensioning of the (e.g. number of) cranes and the construction schedule have been carried out the following model resolves:

$$\text{Minimize } \sum_{t=1}^T \sum_{i=1}^{n_t} \text{cost}_{i,t} \rightarrow \min ,$$

where

T	number of construction stages
n_t	number of existing cranes on the site at the time t
$\text{cost}_{i,t}$	$= \sum \text{cost}_{i,t,e} = \sum \text{dist}(e, p_i) \cdot \# \text{cycles}$
dist	the three-dimensional Euclidean metric
p_i	the center of the floor space of crane i (base of the crane)
#cycles	number of needed crane work cycles

The flow of material belonging to a certain execution element are being considered equally distributed among the four corners of each execution element of the building. Thus, an execution element is not handled as a whole but cornerwise. $\text{Cost}_{i,t}$ is determined by summing up the distances for all the corners the crane i is marked for as a serving crane during construction stage t.

2. IMPLEMENTATION OF THE HEURISTICS

To solve this problem which is described by its input data, its object function, the two constraints and the resulting data, a backtracking algorithm has been designed. This algorithm is working in a way that is comparable to a depth search in the tree structure as it is depicted in Figure 1. It is assumed that a crane is located during the first stage of construction in which

it is needed and that, then, its location will not be changed again. There are T construction stages $(1, \dots, T)$. During construction stage t ($1 \leq t \leq T$) n_t cranes $(1, \dots, n_t)$ have to be located for the first time. During the locating of crane i in construction stage t $m_{t,i}$ grid points (GP) are examined for their suitability for a crane location $(GP_{t,i,1}, \dots, GP_{t,i,m_{t,i}})$.

Using the example of a new locating of crane i in construction stage t ($1 < t < T$, $1 < i < n_t$) the process of finding the crane location shall be demonstrated in reference to Figure 1: it can be assumed that all cranes of construction stages 1 through $(t-1)$ as well as the cranes 1 through $(i-1)$ of construction stage t have been located on the site in accordance with the constraints. In Figure 1 this is represented by the path consisting of nodes with bold borders running from the root of the tree to the node labeled "GP $t,i-1,1$ ".

It is assumed that all of the yet located cranes have been located within the first grid point. In order to be capable of executing the locating of crane i of construction stage t , the data structure containing all areas suitable for the purpose and the suitability ranking numbers of the areas have to be retrieved from the knowledge based system component of the hybrid system⁵. A grid point scheme is put over the suitable areas according to a method which will be described later on. Each grid point (labeled "GP $t,i,1$ ", ..., "GP $t,i,m_{t,i}$ " in Figure 1) is examined for its suitability for the crane location. Also, this method will be described later in this paper. Now the crane will be temporarily located on the most suitable of these points. ("GP $t,i,1$ "). If this locating can be executed in accordance with the constraints and if, also, the cranes $(i+1)$ through n_t of construction stage t as well as all cranes of construction phase $(t+1)$ through T can be located in accordance with the constraints, a possible crane location is found. In this case, a path extends from the root of the tree to a leaf. The nodes of this path contain all the grid points computed by the algorithm for the cranes that have to be newly located during the construction stage. Otherwise, the crane i of construction stage t has to be temporarily located on less suitable points starting at "GP $t,i,2$ " until a possible crane location is found. If the crane has temporarily been located on all grid points without finding a possible location, the location of at least one of the already located cranes has to be changed in accordance with the constraints before a new trial to locate crane i of construction stage t can be carried out.

The order in which the selection of grid points within the suitable areas for the temporary locating of a crane takes place has a major impact on the running time of the algorithm and also for the quality of the generated crane location considering the cost.

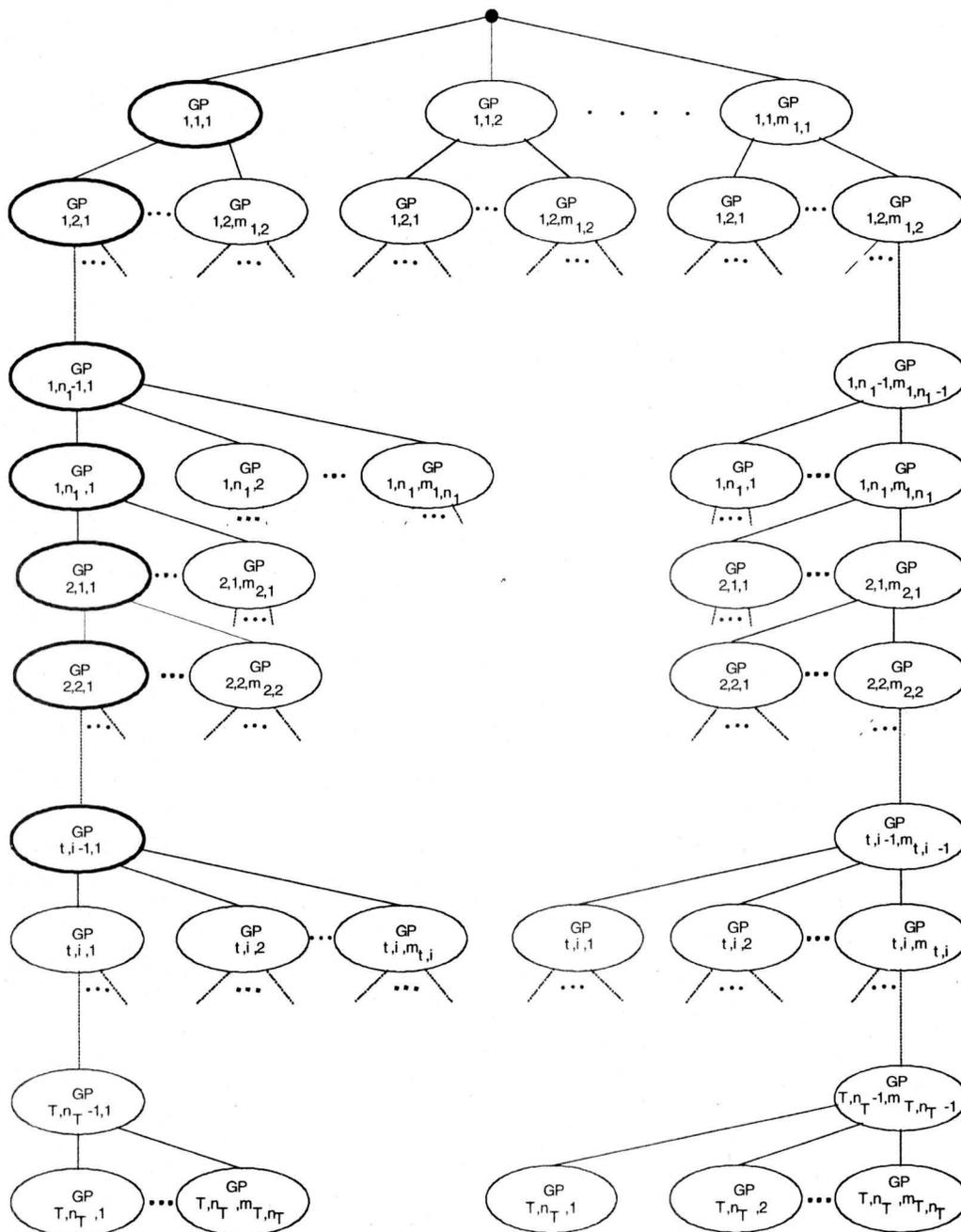


Figure 1 Search tree scheme for the search of a feasible crane layout

2.1 EVALUATION OF THE GRID POINTS

The heuristics used for the evaluation of a grid point takes a look at two quantities:

- first the grid points are sorted according to the number of not yet served corners which the crane could serve during his work on the construction site in case of a locating at these

points. For this purpose all the construction stages are examined in which the crane has to be permanently available on the site. A grid point will be considered superior to a different one if from this grid point more corners can be served than from the other one.

- Grid points which due to this criterion are considered equal will furthermore be assessed according to their related cost. These include the cost arising from the serving of the not yet served corners (according to the above described cost measure). The savings arising from the more cost-effective serving of already served corners by the new crane are subtracted from those cost. If two grid points are equal in terms of their other attributes, the one with lower related costs will be considered superior.

3. AN EXAMPLE FOR THE HEURISTICS

After the behavior of the algorithm has essentially been described, the way it works will be demonstrated by using an example. The data necessary for this originate from the example database in which building 50.31 of the University of Karlsruhe is modeled in a simplified way. Two different cranes will have to be included for all of the 39 construction stages of the construction process. Figure 2 represents this situation. In the center, a plan view on the building can be found. One easily recognizes the four huge floor elements measuring ten meters in breadth and 24 meters in length on which the columns bearing the building are situated.

A grid scheme chosen by the user is put over these. In this case the grid points are usually situated at a distance of 8.5 meters to their next neighbor. Irregularities are due to a changed orientation of the crane ground area caused by closeness to the borders of the suitable areas. In this example we have the unusual case that the suitable areas and the distribution of the grid points are identical for both cranes.

The process of the locating of the cranes for this example can be structured according to this way:

1. In the beginning the first crane will be located on a trial basis. Altogether 62 grid points have to be examined for their suitability for a locating by using the heuristics. Due to the number of corners of execution elements which can additionally be served from this point, the point (54,0; 27,75) is chosen for the first locating of the first crane as a trial.
2. An examination shows that the temporal constraint as well as the weight problem are valid for the first crane in the first construction stage.
3. Now, the second crane has to be located. Also in this case a suitability examination of 62 grid points for the locating of the second crane takes place. Because we have an exceptional case here where the first crane is already registered as the serving crane for all corners of the execution elements, for the second locating it is only important at which grid points the minimum cost are caused or the biggest saving potential can be realized. Therefore, the point (54,0; 27,75) is chosen for the first locating of the second crane on a trial basis.
4. The temporal constraint for the second crane in the first construction stage is valid and so are the spatial constraints - the overlap and weight problem - for the first stage.

5. Because both constraints are valid for the following construction stages (2,.....,39) the two temporary crane locations can be integrated into the feasible crane layout.

The result of the crane layout is demonstrated in Figure 2. Table 1 contains the cost related to the resulting crane layout and the expenditure parameters broken down into cranes and construction stages.

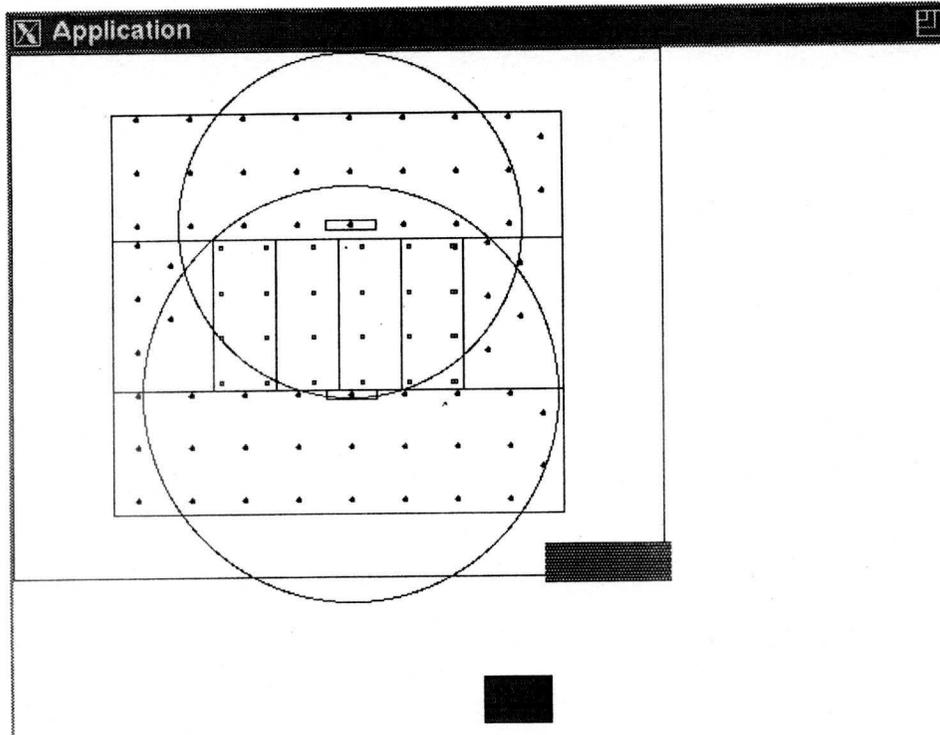


Figure 2 This hardcopy of an EIFFEL session shows the result of the crane layout on the exemplary site; beside the building, the suitable areas and the located cranes one recognizes the border of the site with exit and entrance in the right lower corner; the red button serves for finishing the EIFFEL application

Contrary to what one might conclude from Figure 1, the implementation of the algorithm does not use a structure identical to the depicted search tree. Going through this search tree using backtracking can be simulated by a linear structure (doubly-linked list). "GP $t,i,k_{t,i}$ " stands for the grid point which has just been used for the locating of crane i in construction stage t ($1 \leq t \leq T$, $1 \leq i \leq n_t$, $1 \leq k_{t,i} \leq m_{t,i}$).

CS	crane1				crane2				total cost
	cost	corner	work cycles	dist	cost	corner	work cycles	dist	
1	411	2	24	34	408	2	24	34	819
2	514	2	30	34	510	2	30	34	1024
3	0	0	0	0	0	0	0	0	0
4	182	2	24	15	169	2	24	14	351
5	228	2	30	15	212	2	30	14	440
6	44	2	2	44	43	2	2	43	87
7	166	4	26	42	151	4	26	39	317
8	169	2	30	11	152	2	30	10	321
9	19	2	2	19	17	2	2	17	36
10	332	4	26	41	324	4	26	38	656
.
.
.
30	294	2	30	19	286	2	30	19	580
31	45	2	2	45	45	2	2	45	90
32	227	4	26	49	219	4	26	47	446
33	243	2	30	16	234	2	30	15	477
34	22	2	2	22	20	2	2	20	42
35	366	4	26	48	360	4	26	46	726
36	433	2	30	28	428	2	30	28	861
37	0	0	0	0	0	0	0	0	0
38	26	2	2	26	36	3	3	36	62
39	38	2	2	38	37	2	2	37	75
tot.	8082.05	84	684		7879.91	85	685		15962

Table 1 Result with breaking down of cost and expenditure parameters into cranes and construction stages (CS)

4. A FIRST EVALUATION OF THE CRANE LOCATION ALGORITHM

First, a theoretical estimation of the running time will be delivered. Then, an example is used to give an answer to the question: to what extent does the generated crane location fulfill the demand for low transportation cost?

Altogether n cranes have to be located, i.e. n being $n = \sum_{i=1}^T n_i$ in reference to the notation of

Figure 1. The necessary running time for the locating of these n cranes is

$$t_n = t_n^{XPS} + t_n^{EVAL} + t_n^{REC}.$$

Where

Def.: t_n^{XPS} : Running time of the expert system component of the hybrid system needed for the generation of suitable areas for the location of crane n . In the context of this examination this number has to be considered constant. Therefore, the following is valid:

$$\forall j = 1, \dots, n: t_j^{XPS} = c_{XPS}$$

Def.: t_n^{EVAL} : Running time necessary for the evaluation of the grid points within the suitable areas. The parameters influencing this number will now be described. t_n^{EVAL} surely depends on the number of grid points ($\#GP_n$) which are possible locations for crane n due to the stretching of its standing area. The number of the grid points which have to be examined is anti-proportional to the square of the used grid size when suitable areas are the same. Additionally, t_n^{EVAL} depends on the number of execution elements ($\#EE_n$) of those construction stages in which crane n is permanently located on the site. Therefore, the following is valid:

$$\forall j = 1, \dots, n: t_j^{EVAL} = c_{EVAL} * \#GP_j * \#EE_j$$

Def.: t_n^{REC} : The running time needed for going through the sorted data structure which contains the ranked points and for the recursive descent which is occurring in this connection. This loop has to be gone through until a possible crane layout can be found ($1 \leq i_n \leq \#GP_n$). This requires the expenditure of $t_{serv} = c_{serv} * \#EE_n$. Furthermore, the temporal constraint for the construction stage, in which the crane has to be located again or for the first time, has to be examined. Though the expenditure for this exclusively depends on the number of execution elements, it can be well estimated by using: $t_{temp} = c_{temp} * \#EE_n$. Also, the time needed for locating the remaining (n-1) cranes has a significant impact on the calculation of t_n^{REC} . Therefore, the following is valid:

$$\begin{aligned} \forall j = 1, \dots, n: t_j^{REC} &= i_j * (c_{serv} * \#EE_j + c_{temp} * \#EE_j + t_{j-1}) \\ &= i_j * (c_{REC} * \#EE_n + t_{j-1}) \end{aligned}$$

As a consequence, the time needed for locating n cranes is:

$$t_n = c_{XPS} + c_{EVAL} * \#GP_n * \#EE_n + i_n * (c_{REC} * \#EE_n + t_{n-1})$$

From this can be concluded that t_n heavily depends on i_n . For this reason, the running time behavior will only be examined for two exceptional cases from now on. These cases are the best and the worst case one could think of.

In the worst case, $\forall j = 1, \dots, n: i_j = \#GP_j$ is valid. Then we get:

$$t_n = c_{XPS} + c_{EVAL} * \#GP_n * \#EE_n + \#GP_n * (c_{REC} * \#EE_n + t_{n-1})$$

and after several transformations and insertions we get:

$$t_n = c_{XPS} * \frac{(\#GP)^n - 1}{\#GP - 1} + (c_{EVAL} + c_{REC}) * \#EE * \left(\frac{(\#GP)^{n+1} - 1}{\#GP - 1} - 1 \right)$$

Thus, using the O-calculus for asymptotic running time analysis the following can be said: in the worst case $t_n \in O((\#GP)^n * \#EE)$ is valid. The maximum running time is growing linearly with the number of the considered execution elements, with n cranes it is growing

according to the power of n of the number of grid points which have to be examined and it is growing exponentially with the number of cranes which have to be located. A running time behavior of this kind could prove unbearable even with smaller sites.

In the best case $\forall j = 1, \dots, n: i_j = 1$ is valid. In this case we get:

$$t_n = c_{XPS} + c_{EVAL} * \#GP_n * \#EE_n + c_{REC} * \#EE_n + t_{n-1}$$

and after several transformations and insertions results:

$$t_n = n * c_{XPS} + n * \#GP * \#EE * c_{EVAL} + n * \#EE * c_{REC}$$

Thus, in this case $t_n \in O(n * \#GP * \#EE)$ is valid. The running time in the best case is growing linearly with the three parameters n , $\#GP$ and $\#EE$.

Against the background of the considerable differences in running time between the best case and the worst one, it is of significant importance for the practical usability of the algorithm that a possible crane location is found very fast and that the necessity for a backtracking during the planning process only occurs occasionally or never. In the example this was the case because of the chosen heuristics. The running time amounted to eight minutes on a DECstation (including database requests etc.).

Besides the needed running time the quality of the generated crane location considering cost aspects is another important criterion for the evaluation of the algorithm. To find a first answer to this question all $62^2 = 3844$ combinations (62 grid points, grid size 8.5m) for the locations of the two cranes were examined by using an exact method.

The examination of the 3844 possible locations for the cranes showed that only at about 5% of the examined combinations (192) were a feasible crane location. The other ones were exclusively rejected for not fulfilling the spatial constraint. No possible location was rejected for the temporal constraints.

To the greatest possible extent the possible combinations can be assigned to the grid points which have been considered rather favorable for the locating of the first crane by the heuristics. In case the first crane is regarded as being permanently located, a possible crane location is more likely to be found if the second crane is located at the points which seem to be favorable according to the heuristics.

Figure 3 which shows an illustration following a Lorenz curve, underlines that the heuristics is capable of determining a feasible crane location. One clearly recognizes the accumulation of feasible crane locations at the grid points which are more suitable for the locating of the first crane.

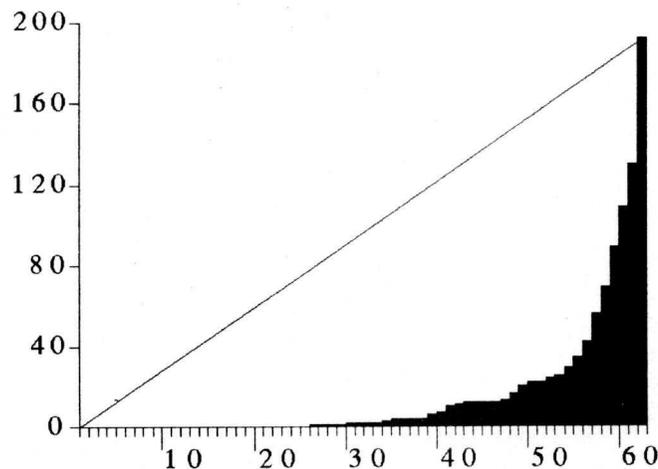


Figure 3 Representation following the Lorenz curve. It shows the distribution of possible crane locations to the grid points sorted according to their suitability for the locating of the first crane

CONCLUSION

If we take a look at the solution of our example, it becomes obvious that no already made decisions to locate a crane in one grid point had to be withdrawn in the process of backtracking and, therefore, the algorithm was able to find a feasible crane location in the fastest possible way. Furthermore, the solution found is optimal in terms of the costs when the examination is restricted to the considered grid points and thus it is an optimum solution. As a matter of fact, the use of this algorithm on other, bigger construction sites on a trial basis seems to be very promising. Right now we improve the algorithm by using better input data from the knowledge-based component. Through the use of better rectangles with a given suitability factor for each facility (e.g. crane), it will be possible for the algorithm to find „optimal“ feasible locations faster.

REFERENCES

- [1] Rodriguez-Ramos, W.E.: Quantitative Techniques for Construction Site Layout Facilities, Dissertation, University of Florida, 1982
- [2] Warszawski, A., Peer, S.: Optimizing the Location of Facilities on a Building Site, Operational Research Quarterly, Vol. 24, No. 1, 1973, S. 35-43
- [3] Warszawski, A.: Multidimensional Location Problems, Operational Research Quarterly, Vol. 24, No. 2, 1973, S. 165-181
- [4] Lennerts, K., Gehbauer, F., L. Petersson: Evaluation of three different methods concerning the crane coverage problem, Proceedings of ECPPM '94-The first European Conference on Product and Process Modelling in the Building Industry, Dresden, Germany, 1994
- [5] Lennerts, K., Gehbauer, F.: A Knowledge Support Component for a Site Optimisation Algorithm, Vol. 72 „Knowledge Support Systems in Civil Engineering“ IABSE-Colloquium Report, Bergamo, Italy, 1995