LogForum

2012, 8 (3), 177-189

> Scientific Journal of Logistics <

http://www.logforum.net p-ISSN 1895-2038 e-ISSN 1734-459X

METHODS OF USING THE QUADRATIC ASSIGNMENT PROBLEM SOLUTION

Izabela Kudelska

Poznan University of Technology, Poznań, **Poland**

ABSTRACT. Background: Quadratic assignment problem (QAP) is one of the most interesting of combinatorial optimization. Was presented by Koopman and Beckamanna in 1957, as a mathematical model of the location of indivisible tasks. This problem belongs to the class NP-hard issues. This forces the application to the solution already approximate methods for tasks with a small size (over 30).
Even though it is much harder than other combinatorial optimization problems, it enjoys wide interest because it models the important class of decision problems.
Material and methods: The discussion was an artificial intelligence tool that allowed to solve the problem QAP, among others are: genetic algorithms, Tabu Search, Branch and Bound.
Results and conclusions: QAP did not arise directly as a model for certain actions, but he found its application in many areas. Examples of applications of the problem is: arrangement of buildings on the campus of the university, layout design of electronic components in systems with large scale integration (VLSI), design a hospital, arrangement of keys on the keyboard.

Key words: QAP, genetic algorithms, Branch and Bound, Tabu Search.

INTRODUCTION

The idea of optimum resource management is connected with determining proper relations between these resources and objectives. The optimum resource management relates with the establishing proper relations between determined resources and objectives. Occurring dependencies determine the efficiency and effectiveness and thus provide optimization. Therefore, the author presents possibilities of applying the method of Quadratic Assignment Problem (QAP).

The QAP problem is one of most interesting and most difficult problems from the area of combinatorics that appear in practice. This problem designs the issue of the allocation of the set of stores for the set of objectives, taking into account the knowledge of the size of costs of assignment and the tendency to minimize this cost. There have been many areas, in which the problem can be applied:

- ergonomics,
- architecture,
- computing science,
- logistics (distribution planning) and production.

The first chapter presents the essence of the problem of the Quadratic Assignment Problem. The author describes the mathematic model prepared by authors of the idea. Next part illustrates the

Citation: Kudelska I., 2012, Methods of using the Quadratic Assignment Problem Solution. LogForum 8 (3),177-189
URL: http://www.logforum.net/vol8/issue3/no1
Accepted: 01.06.2012, on-line: 30.06.2012.

calculative complexity of the QAP, bottom limits for costs, which constitute the basis for determining the algorithm of optimum solutions. Next, most frequent heuristics used for QAP solutions will be presented. In the summary, the author presents applications of the QAP problem.

DESCRIPTION OF THE QAP PROBLEM

Before discussing the QAP problem, the author wants to explain the question of the Problem Linear Assignment (LAP). The explanation of principal differences between these problems can help to understand better the definition and idea of QAP.

Linear Assignment Problem

The common application of LAP used by Hanan and Kurtzberg is known as assigning *n* people to *n* work. Each allocation is related to a certain cost – c_{ij}, assigning an *i* person to *j* work. The assigning of every person is made in order to assign him/her to only one workstation in a way to minimize the total of costs for each task, i.e. its total cost. The problem can be formed in a following mathematic way:

$$\min \sum_{i=1}^{n} c_{i\pi(i)} \qquad (1)$$

It is a set of permutation *1, 2, …, n* i $j=\pi(i)$ for all permutations possible $\pi=S_n$; $S_{n.}$ Every set of tasks is a set of permuations for *n* numbers, and therefore it is equal *n!* we might observe that in case of high values of *n* we are not able to verify all combinations, or such verification is very difficult. For example: if we try to assign *n=10* people to 10 workstations in a way, that has been described above, then we should verify *10!* i.e. about 3.63 milion different combinations.

Quadratic Assignment Problem – Mathematic Model

Our target is a more complicated generalization of the Linear Assignment Problem. Beside the matrix of costs – like in LAP – it encloses also a matrix of distance participation. In order to maintain the cohesion, we might refer to works of Hanan and Kurtzberg [8] and their interpretations of the QAP. They used the case of allocation of offices for people. We find many mathematic definitions of the problem in the literature.

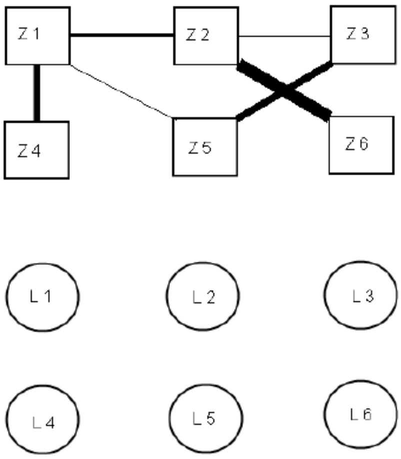
In QAP we obtain a matrix of costs $C = [c_{ij}]$, in which c_{ij} is the measure of the relation between the person *i* and the person *j*. we take under consideration *n* possible offices, to which people may be assigned. This gives us finally a so-called distance matrix $D = [d_{kl}]$, in which D_{kl} means the distance between the office *k* and the office *l*. let's assume that the *i* person is assigned to the *p(i)* office and that the *j* person is assigned to the *p(j)* office. Then, the cost related to the task is being treated as $c_{ij}d_{p\,(i)\,p\,(j)}$. And so the total cost of all tasks of assigning the office will be a sum of particular $c_{ij}d_{p(i)p(j)}$ in relation to all *i,j*. the optimum solution of assigning will take place only when the total cost will reach its minimum.

Similarly to the case of LAP, where there are *n!* permutations , from which one can choose the optimum assigning. However, here, the principal difference between these two problems is that in contrast with LAP, in which the assigning of *j* work to a *i* person was made independently from tasks of other workers, in QAP tasks are not independent from each other. This means that one must take under consideration the assigning for everyone, who has a non-zero relation with an individual in the study of assigning this particular person *i*.

The Quadratic Assigning Problem can be formed as follows:

There are *n* localizations and *n* tasks given. We also know a flow matrix $A = a_{ij}$, in which a_{ij} is for example a number of materials flowing from the task *i* and the task *j* during one unit of time. The cost matrix is $B = b_{ij}$, where b_{ij} is the distance between the localization *i* and the localization *j*. the cost of assigning the task $\varphi(i)$ to the localization *i*, and the task $\varphi(j)$ to the localization *j* is equal $a_{\varphi(i)\varphi(j)}b_{ij}$. the assigning of all tasks can be presented in a form of permutation *' 2 Sn.*

The figure 1 presents the problem in a graphic form. On this figure, Z1, Z2, ..., Z6 mean tasks, the flow between these tasks (matrix A) is marked by segments of different thickness – this symbolizes different flows between tasks. The problem concerns placing tasks Z1,Z2,..., Z6 in proper localizations in a way to minimize the total cost of assignment $a_{\varphi(i)\varphi(j)}b_{ij}$.



Source: Burkard, Dell'Amico, Martello, 2008

Fig. 1. Quadratic Assignment Problem
Rys. 1. Kwadratowy Problem Przydziału

The research in the literature of the problem shows that there are many mathematical definitions of the issue mentioned before. The author presents also the definition created by Koopman and Beckmann in 1957.

Koopman – Beckmann assumed that there are following matrixes:

$A = (a_{ik})$ – in which a_{ik} means the flow between the task *i* and the task *k*;
$B = (b_{jl})$ – in which b_{jl} is the distance between the localization *j* and the localization *l*;
$C = (c_{ij})$ – in which c_{ij} is the cost of placing the task *i* in the localization *j*;

Then, the problem has the form as presented in the figure 1 and in the formula (2)

$$\min_{\varphi \in S_n} \left(\sum_{i=1}^{n} \sum_{k=1}^{n} a_{ik} b_{\varphi(1)\varphi(k)} + \sum_{i=1}^{n} c_{i\varphi(i)} \right) \tag{2}$$

In which S_n is the permutation of natural numbers 1, 2, 3, …, n.

The product $a_{ik}\, b_{\varphi(i)\varphi(k)}$ is the cost of combining the task *i* with the localization $\varphi(i)$, and the task *k* with the localization $\varphi(k)$. Each element from the set of conditions $c_{i\varphi(i)} + \Sigma^{n}_{k=1}\, a_{ik}\, b_{\varphi(i)\varphi(k)}$ is a total cost of assigning the task *i* to the localization$\varphi(i)$, and every task *k* to following remaining localization$\varphi(1),\varphi(2), ..., \varphi(n)$. The problem described this way, along with set matrixes *A,B,C* that we write *QAP*(*A,B,C*). if the problem has a non-linear character (without the known matrix *C*), we write it in a form *QAP*(*A,B*)[1].

The problem has been applied in the issue of assigning economic activities to the localization, taking under consideration the quadratic quality indicator. It remains until today its main application.

It has become popular because it generates many problems from the area of combinatorics, which can be faced in problems of the real practice.

HEURISTICS AND ACCURATE METHODS

In Linear Assignment Problems it is applied among other the Hungarian Algorithm. The cost matrix encloses in its columns tasks, in its rows - means that are in hand. Each cell refers to a cost of assigning means to a determined task. A cell in the first row and first column is the cost of assigning the first asset to the first task. This method fulfills its objective accurately for Linear Assignment Problems. However, assignment problems with square cost coefficient is a completely other problem. Accurate methods are not applied in cases bigger than 25 because of their big computational complexity. In such cases, when a big project is studied, it is enough to obtain an approximate solution; then it is possible to apply heuristic methods, which the author will present in the following part of the elaboration.

From all accurate methods, best results for difficult problems come from the Branch and Bound algorithm.

The exceptional difficulty of QAP has caused the development of heuristic methods of research. Genetic Algorithm, Tabu Search, Simulated annealing, and other specialistic methods, have been applied for solving QAP. The efficiency of various heuristics differs from each other with certain features, characteristic for every individual method. [P.Ji, 2006, p.108]

Branch and Bound

The algorithm is basing its action on the division of the issue on many smaller and less complicated ones. It is possible thanks to the search tree structure. Each branch of such tree represents a "sub-issue" that occurred in result of a division of an issue from a higher level. On the consecutive level of the tree incurred branches are being compared. This allows rejecting the worse branches, which results minimizing the amount of following sub-issues for studying. It is very important because the necessity of verifying all possible solutions doesn't make sense from the practical point of view, if the number of solutions is bigger than 25. The method is frequently used for small and medium problems.

The method of isolated assignment is the simplest and the most frequently applied method (Gilmore, Lawler). It is based on putting not-assigned objects to next locations. Thanks to such proceedings, one is creating an issue, which is the sub-problem of the issue of the higher level.

It is not the only method possible to find in the literature. Gavett and Plyter suggested an algorithm for pairs assignment. It is based on assigning a pair of tasks to a pair of localizations. Mirchandani and Obata present another approach; they propose a relative positioning algorithm. The idea of Roucairol is another solution; he named his algorithm Polytomic branching. The next step is to choose the best path for further search, which is being found by finding the lower limit. The GLB technique (Gilmore-Lawler bound) is very popular; its main advantage is that it's calculative side is not complicated.

One knows trials of upgrading the division and limitation method made by applying techniques for finding bottom limits. Hahn, Grant, and Hall presented a technique called Dual Procedure (DP).

Genetic Algorithm

Starting from the sixties of the 20th century, numerous scientific centers formed all sorts of schools promoting methodologies and algorithms of heuristic searching developed by oneself; methodologies, which were often inspired by mechanisms observed in the natural evolution. In the beginning of the nineties, representatives of all these schools met on a conference and they agreed that they shared the same object of the interest, which were evolulutionary computation. Since then, there has been accepted to name all originally developed methods and their further alterations, which ofter result from hybridization, with the name of "evolutionary algorithms". Primitive divisions didn't disappear entirely and until today disputes and controversies continue between representatives of various directions. The group of evolutionary algorithms encloses particularly genetic algorithms and evolutionary strategies. Some authors also add to this group the method called dispersed searching, although this heuristic differs creatively from previous in a quite significant way [Grygiel, 2011] [Łęski, 2008].

The genetic algorithm has been presented for the first time in 1975 by Holland. Its characteristic is that the algorithm has a binar representation of individuals and recombination with use of the single-point crossing, the mutation of bits of representatives and the reproduction with use of the roulette method (proportional) [Łęski, 2008].

The primary version of the genetic algorithm is currently called elementary, canonical, or simple genetic algorithm [Grygiel, 2011, Rutkowska and others, 1997].

The essence of action of genetic algorithms consists in searching the space of solutions for the problem, in order to find best solutions (taking into account a determined criterion). The algorithm operates on a certain population of individuals, i.e. solutions. This allows to search the space of solutions from different points of view in the same time. The essential genetic algorithm is made by following steps [Tate, 1995, Rutkowska and others, 1997, Łęski, 2008]:

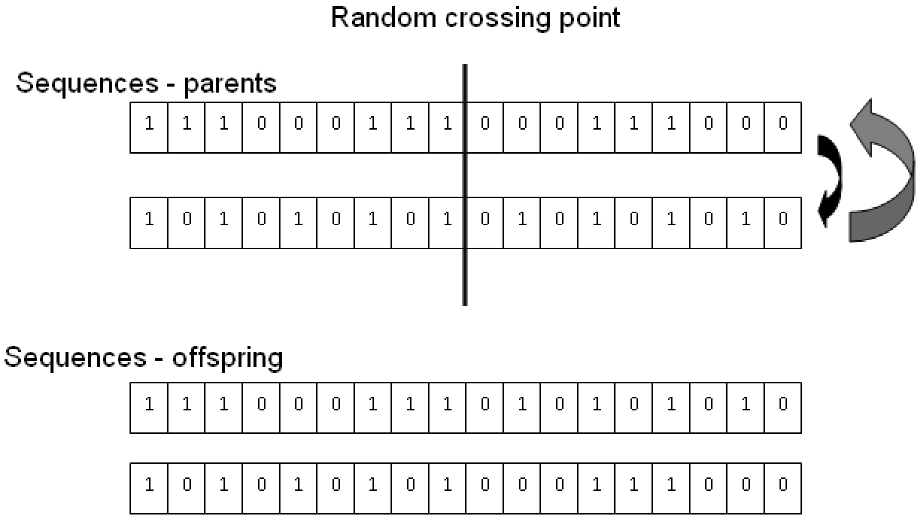
- The imitation i. e. choice of the initial population of chromosomes,
- Evaluation of the range of adaptation of chromosomes to the population,
- Verification of retention conditions,
- Selection of chromosomes,
- Application of genetic operators,
- Formation of the new population,
- Modelling of the "best" chromosome.

In the beginning of the work with use of the genetic algorithm there always must be created an initial population of code sequences - the first generation, which can be presented in form of the following formula (formula 3):

$$P(a) = \{x_1^a, x_2^a, \dots, x_b^a\} \qquad (3)$$

The set of these sequences can be made in an absolutely random way; however its formation can also be based on certain rules. A set of coded sequences allows to calculate a suitable set of values for functions of objective in the same time. Classical genetic algorithms work with coded sequences written in form of chains of zeros and ones; however it is possible to write a coded sequence with use of a bigger number of letters of the alphabet. Still, the most important thing is to have the possibility of decode essential information from such sequence [Knosala 2002]. Detailed information on the process of coding can be found in the following positions: [Sarker and others 2002, Anderson 2002, Rutkowska and others 1996].

Each sequence of codes is a properly coded data concerning the required value of variables, for which the value of variables should reach the extreme value. Usually, just before the formation of the first generation, one introduces parameters for genetic algorithm work into the program: number of generations, probability of mutation and crossing. After creating the first generation, each coded sequence must be calculated with use of the function of the objective prepared for this sequence. Then, after determining values of all sequences, the draw (selection) to so-called intermediary generation, takes place. The generation has the same size, i.e. the same number of sequences, as the first generation; however it doesn't have to be always this way. Sequences evaluated as "better" have more chances for entering into the determined generation and being copied into it. The chromosomes selection is about choosing on basis of calculated values of functions of chromosomes adjustment, which of them will participate in the formation of the next generation representatives. There are many methods of selection. The most popular is the so-called roulette method. When the intermediary generation is already formed, sequences of this generation are being subjected to the operation of crossing. In the simplest case it concern a random selection of a pair of sequences, a random selection of the crossing point inside sequences and exchange of places of fragments of sequences (figure 2) [Knosala 2002, Goldberg 1995, Rutkowska and others 1997].



Source: Personal elaboration based on [Knosala, 2002, p. 401]

Fig. 2. Sequence crossing operation
Rys. 2. Operacja krzyżowania ciągów

The literature describes many various crossing operators, which function in different ways. More information on the subject can be found in following positions: [Goldberg 1995, Michalewicz 1996, Sarker and others 2002, Rutkowska and others 1997]. However, the main goal is always to exchange information between sequences in a way to use best characteristics of parent sequences in order to form even better sequences - children [Knosala 2002, Rutkowska and others 1997, Michalewicz 1996].

After having formed the required number of children, the mutation is made. It concerns a random exchange of representation bits with small probability. Similarly to the reproduction, the mutation operates on individual parental representative. And similarly to the case of crossing, there are many types of mutation operators. However, the objective of applying the operator is always the same: to introduce the modification into sequences in order to cause the initiation of searching the space of solutions with use of the genetic algorithm in new areas, that haven't been studies yet [Knosala 2002]

Apart from two principal genetic operators, there are also a big number of other operators, for example compression [Michalewicz 1996, Goldberg 1995, Knosala 2002].

When all necessary operations on coded sequences are made, the newly formed set of sequences creates an absolutely new generation. The work of the algorithm can be finished, when the mean of the function of the objective next generations do not differ from each other [Knosala 2002].

Because of the essence of its functioning, each genetic algorithm for every individual problem must enclose following elements [Michalewicz 1996, Arabas 2001, Cytowski 1996]:

- Principal representation of potential solutions of the problem,
- Method of creating the initial population of potential solutions,
- Function of evaluation, which has the role of the environment and it assesses solutions,
- Principal operators that affect the composition of the children-population,
- Values of different parameters used in the genetic algorithm (the size of the population of the probability of using genetic operators)

The basic difference between genetic algorithms and other techniques lays in the view on the process of evolution. Genetic algorithms focus on the evolution of individual units, while other techniques stress the importance of the population's evolution out.

Simple genetic algorithms don't provide solution to certain sort of problems. This involves among others the constant length of the DNA chain. Therefore, there have been introduced an extension of the genetic algorithm, which was named ESGA.

Genetic algorithms have both strong and weak sides. Their advantages are as follows [Haupt 2004]:

- they optimize with constant and discrete variables,
- they do not require derivative of the information,
- they simultaneously search a big sample,
- they handle a big number of variables,
- they optimize composite variables,
- they provide a list of optimum variables instead of individual solutions.

From the other side, their disadvantages are:

- lack of sensible criteria of stoppage,
- calculative complexity,
- we are never certain whether the solution we found is optimal.

Simulated annealing

The simulated annealing is another method for optimization. It was presented by N. Metropolis (co-originator of MANIAC and MANIAC II computers and one of authors of Monte Carlo methods) and others in 1953 and remained later by S. Kirkpatrick and others - in 1983. The method develops iterative methods, which were based on continuous upgrading the existing solution, until it reached the moment, when the solution could not be improved anymore [Łęski 2008].

The idea of the simulated annealing method comes from the statistical physics. It is based on the analogy the behaviour of materials, which are processes with a temperature that falls with a determined tempo. If freezing out is slow, the material aims at reducing its state of energy. However, if the freezing is fast, the material transfers into an amorphous structure, with energy state that is higher than previously. In other words: molecules of a liquid move freely in a high temperature. However, when the temperature falls, molecules begin to move the more and more slowly, and gradually they form an ordered structure - a crystal. This state is characterized with a minimum level of energy. Slow cooling down the system is the necessary condition for forming the cristalic liquid; otherwise molecules of the substance will not find the optimum localization and they will form themselves in a more chaotic structure [Metropolis and others 1953, Kirkpatrick and others 1983, Łęski 2008]

The basic formula used in termodynamics for describing the phenomenon presented above, which has been transferred to the described algorithm is as follows (formula 4):

$$P(E) \approx e^{\left(\frac{-E}{kT}\right)} \qquad (4)$$

where:

k is the Boltzman constant.

The minimized function of the criterion is an analogy of the state of energy of the material.

Physics quantity of the temperature is being replaced by a so-called *pseudotemperature*. In the simulated annealing, in the situation, when the material is being cooled down, its energy is generally reduced; however its temporal increase is accepted in order to leave the local minimum [Łęski 2008].

The method can be presented in a following way [Łęski 2008]:

1. We choose the initial value θ and the initial pseudotemperature T;
2. We choose Δθ on basis of the probability distribution given by a so-called *generation g function* (Δθ, T);
3. We calculate ΔJ = J(θ + Δθ) – J θ;
4. We accept θ ⟵ θ + Δθ with the probability given by a so-called *acceptation* α *function* (ΔJ, T);
5. If there were made the assumed number of iterations for the temperature T, then we reduce the pseudotemperature T ⟵ λT, where λ ∈ (0, 1);
6. If conditions of the alloy are not fulfilled, then we pass to 2); otherwise - stop.

The operation from the step 5, which is supposed to reduce the pseudotemperature, is called the *annealing schedule*.

The parameter of the algorithm which is affecting the probability of worse choice of the answer is the parameter transferred directly from thermodynamic basics of the algorithm, i.e. the temperature. In the beginning of the operation, the temperature is high, thanks to this fact the algorithm can change the solution configuration very often, choosing worse solutions frequently. Along with next iterations of the algorithm the temperature is dropping and better solutions are selected more and more often. In the end of the algorithm work the temperature is so low that the probability of making the worse choice is nearly equal none. Then, the algorithm behaves in the same way as a typical iterative algorithm and it tries to improve the solution to the maximum.

There is a type of Boltzman machine – called the Cauchy machine or, in other words, the algorithm of fast simulated annealing, in which the function of the generation is determined as (formula 5) [Szu and others 1987]:

$$g(\Delta\theta, T) = \frac{T}{\left(\|\Delta\theta\|^2 + T^2\right)^{\frac{p+1}{2}}} \qquad (5)$$

The literature presents it also as a so – called very fast simulated annealing.

In many practical situations the application of simulated annealing is not possible because of the computational effort related to the required annealing schedule. It is possible to present it in other words: in order to obtain the global minimum, it is necessary to reduce the pseudotemperature very slowly.

Tabu Search

The Tabu Search Method has been accepted by Glover and Hansen to be an efficient technique for solving composite problems of optimization. Just like genetic algorithms, the determined method has the ability to lead the method to the local searching, in order to avoid the weak local optimum.

The process of searching the space of solutions is being coordinated with strategies based on mechanisms of the memory, which characterize the TS algorithm.

Principal elements of the Tabu Search are [Bożejko and others 2010]:

- The environment - the sub-set of acceptable solutions;
- The motion - the function that transforms one solution into another;
- The Tabu List - the list that contains attributes of a certain number of solutions in question.

The main idea of the Tabu Search is to use the memory and remember solutions ore movements (changes). Along the course of search, it gathers information on researched space. Local choices depend on information gathered during the entire search. Basing on information recorded in the memory it forms limitations that protect the algorithm from coming back to areas of the space that have been already searched. These limitations depend from following aspects:

- The frequency of recording determined data,
- The timeliness of data recorded in the memory,
- The impact of determined data on the quality of results obtained in the method.

In its following interactions, the algorithm searches the nearest area around the solution that has been found - in order to determine the new localization of the current solution. Therefore, it has to be defined the relation of the neighborhood for all elements of the determined space, in which we search solutions.

The memory structure stores information concerning all realized transformations. It might contain also other information, like for example: concerning the frequency of these transitions in time that has passed after the realization of one of these transitions. There are two types of memory in TS algorithms: the sort-term memory (The short-term memory is being exploited in every iteration and it serves to remember solutions that have been visited lately; its main task is to prevent choosing the movement operator that might lead to the formation of a loop in the algorithm and blocking it in a very small space of research [Glover, Laguna]) and the long-term memory (It stores information on the course of the process of searching, it allows to remember best solutions from searched areas of the space, instead of keeping only the best solution from the current neighborhood [Glover, Laguna]). Each of them is used by strategies, which are characteristic for them. Effects of their functioning can be seen in the form of the modification of the neighborhood N(x) for the current solution x. The modified neighborhood N*(x) is the result of storing information concerning the process of searching that already took place [Glover, Laguna, Witczak 2010]

The Tabu Search Algorithm realizes fully the process of searching in the area of the nearest neighborhood of the current solution. The present solution is being replaced by a best solution in the neighborhood, even if this may cause the decrease of the level of quality. The process of searching uses the system of limitations put on the set remaining the part of the neighborhood. It is supposed to prevent the occurrence of the possibility of forming a loop in the algorithm and returning to the same, narrow areas of solutions. Solutions which earlier were accepted as current answers are being removed from the defined neighborhood. This way, a so-called tabu set is being created. Limitations used in the Tabu Search Method have the form of an absolute interdiction or certain restrictions.

The author present below an exemplary procedure of the Tabu Search [Komosiński 2010]:

The Tabu Search Procedure:

```
begin
    INITIATE (zstart, xbest, T)
    x := xstart
    repeat
        GENERATE ($V \subset N(x)$)
        SELECT (x’) // best $f$ in V + aspiration
        UPDATE TABU_LIST(T)
        if $f(x') \leq f(xbest)$ then xbest := x’
        x := x’
    until STOP CONDITION
end
```

It has been accepted to use following symbols:
T – tabu set,
V – list of candidates,
N(x) – neighbourhood,
f(x’) – movement quality,
x – initial solutions,
xbest – accepting the solution x the be the best solution,

In the presented procedure we deal with the selection of the initial solution x, and next - with accepting it to be the best solution. In the set of candidates from the neighborhood we choose the best one, taking into account the tabu information. Then, we update the structure of the memory.

Methods presented in this chapter have many various types of the extension and alterations, which could constitute a subject of individual chapters. Presented information concerning heuristics is only very general and it is described in order to bring the problem closer to the reader. However, those, who are interested in the question, are invited by the author to check the literature and websites enumerated in the references.

SUMMARY

The QAP issue can be used in multiple real situations. One of first examples of its application was placing buildings in the university campus. This application has been described with details by J. W. Dickey and J. W. Hopkins. It has been named "campus planning model" and it was presented in the work Campus building arrangement using 8 TOPAZ in the year 1972. Its aim was to reduce the distance between buildings.

Next similar applications were: park planning (J. Bos Enviromental Management 1993), and the project of a hospital (Elshafei 1977). From the today's point of view, this problem can only be called historical. At present we use QAP for absolutely different and more complicated problems.

Issues from the reality, just like other methods of solving problems, also evolved. The problem of placing buildings in a campus has been replaced by the problem of placing keys on the keyboard or on the steering panel. The minimization of distance between most frequently used keys is the objective - this results with shorter time needed for writing a text. The problem has been presented in a model by M. A. Pollatschek, N. Gershoni and Y. T. Radday in their work Angewandte Informatik, presented in 1976. There are various similar evolutions of this problem, for example: localization of indicators on the steering board. In such case, the minimization encloses the distance between the most important and mutually related indicators.

Also the turbine runner problem has a very interesting application. Balancing the turbine of the engine is nature of this problem. Blades of the jet engine have different mass, these differences occur in the stage of production. The centre of gravity must agree with the pivot of the turbine. J. Mosevich used the QAP problem for this case and he has presented it in his work Balancong.

The QAP problem is still very difficult. Further progress and development in these areas will certainly lead to solutions for even more complex cases. Today, the majority of best solutions for large authorities in QAPLIB is obtained through metaheuristics.

REFERENCES

Anderson P., 2002, Introduction to Genetic Algorithms, Rochester Institute of Technology, Lecture, December 15

Arabas J., 2001, Wykłady z algorytmów ewolucyjnych [Lectures on evolutionary algorithms], WNT, Warszawa

Bartolomei-Suárez S., Egbelu Pius J., 2000, Quadratic assignment problem QAP with adaptable material handling, International Journal of Production Research, Vol. 38, No. 4, Taylor & Francis, pp 855-873

Burkard R., Çela E., Karisch S. E., Rendl F., 2006, QAPLIB - A Quadratic Assignment Problem Library, http://www.seas.upenn.edu/qaplib/ access: 19.07.2011r.

Burkard R., Dell'Amico M., Martello S., 2009, Assignment Problems, Society for Industrial and Applied Mathematics, Philadelphia

Çela E., 1998, The Quadratic Assignment Problem Theory and Algorithms, Kluwer Academic Publishers, Boston/London

Christofides Nicos, Benavent Enrique, 1989, An Exact Algorithm for the Quadratic Assignment Problem on a tree, Operations Research, Vol. 37, No. 5, September - October

Commander C. W., 2005, A Survey of the Quadratic Assignment Problem, with Applications, Morehead Electronic Journal of Applicable Mathematics, Issue 4,

Cytowski J., 1996, Algorytmy genetyczne. Podstawy i zastosowania. Problemy współczesnej nauki. Teoria i zastosowania[Genetic algorithms. Fundamentals and applications. The problem of modern science. Theory and applications], Akademicka Oficyna Wydawnicza PLJ, Warszawa

Geoffrion A. M., Graves G. W., 1976, Scheduling Parallel Production Lines with Changeover Costs : Practical Application of a Quadratic Assignment / LP Approach, Operations Research, Vol. 24, No. 4, July - August

Gilmore P.C., 1962, Optimal and suboptimal algorithms for the quadratic assignment problem, SIAM Journal on Applied Mathematics 10, pp. 305 - 313

Glover F., Laguna E., Taillard E., de Werra D., (eds.), 1993, Tabu search, Annals of Operations Research 41

Goldberg D. E., 2003, Algorytmy genetyczne i ich zastosowania [Genetic algorithms and their applications], Wydawnictwa Naukowo Techniczne, Warszawa

Haupt R. L., Haupt S. E., 2004, Practical Genetic Algorithms, John Wiley&Sons, New Jersey

James T., Rego C., Glover F., 2009, Multistart Tabu Search and Diversification Strategies for the Quadratic Assignment Problem, IEEE Transactions on Systems, MAN, and Cybernetics, part A, Systems and Humans, Vol. 39, No 3

Ji P., Yongzhong Wu, Haozhao Liu, 2006, A Solution Method for the Quadratic Assignment Problem (QAP), The Sixth International Symposium on Operations Research and Its Applications (ISORA '06), pp. 106 - 117

Juszkiewicz M., 2010, Zastosowanie algorytmu pszczelego w rozwiązaniu zagadnienia kwadratowego przydziału[The use of bees algorithm to solve quadratic assignment problems], Kraków

Kirkpatrick S., Gelatt C., Vecchi M., 1983, Optimization by simulated annealing, [in:] Science, 220, p. 671 - 680

Knosala R., 2002, Zastosowania metod sztucznej inteligencji w inżynierii produkcji [Applications of artificial intelligence in manufacturing engineering], Wydawnictwa Naukowo Techniczne, Warszawa

Komosiński M., 2010, Optymalizacja: przeszukiwanie tabu [Optimization: Tabu Search], http://www.cs.put.poznan.pl/mkomosinski, access: 19.06.2011 r.

Łęski J., 2008, Systemy neuronowo rozmyte [Neuro fuzzy systems], Wydawnictwa Naukowo Techniczne, Warszawa

Metropolis N., Rosenbluth A. W., Rosenbluth N. M., Teller A. H., Teller E., 1953, Equation of state calculations by fast computing machines, [in:] Journal of Chemical Physics, 21 (6), p. 1087 - 1092

Michalewicz Zb., 1996, Algorytmy genetyczne + struktury danych = programy ewolucyjne [Genetic Algorithms + Data Structures = Programs Evolutionary], Wydawnictwa Naukowo Techniczne, Warszawa

Misevicius A., 2005, A Tabu Search Algorithm for the Quadratic Assignment Problem, Computational Optimization and Applications 30, pp. 96 - 111

Morin T. L., Marsten R. E., 1976, Branch and Bound Strategies for Dynamic Programming, Operations Research, Vol. 24, No. 4, July - August,

Povh J., 2008, Assignment Problems in Logistics, Logistics & Sustainable Transport, Vol. 1, Issue 3

Rutkowska D., Piliński M., Rutkowski L., 1997, Sieci neuronowe, algorytmy genetyczne i systemy rozmyte [Neural Networks, Genetic Algorithms and fuzzy systems], PWN, Warszawa

Rutkowski L., 2006, Metody i techniki sztucznej inteligencji [Methods and techniques of artificial intelligence], Wydawnictwo Naukowe PWN, Warszawa

Sarker R., Newton C., 2002, A genetic algorithm for solving economic lot size scheduling problem, Computers and Industrial Engineering 42, p. 189 - 198

Stützle T., 2006, Iterative local search for the quadratic assignment problem, Eur. J. Operations Research 174, pp. 1519 - 1539

Tseng L., Liang S., 2006, A hybrid metaheuristic for the quadratic assignment problem, Comp. Opt. Appl. 34, pp. 85 - 113

Witczak E., 2010, Zastosowanie algorytmów poszukiwania z tabu do optymalizacji układania planu zajęć [Application of tabu search algorithms for optimization scheduling], [in:] Studia i Materiały Informatyki Stosowanej [Studies and applied informatics materials], tom 2, nr 2, Bydgoszcz, p. 59 - 66

Zhu W., Curry J., Marquez A., 2010, SIMD Tabu Search for the Qauadratic Assignment Problem with graphics hardware acceleration, International Journal of Production Research, Vol. 48, No. 4, pp. 1035 - 1047.

METODY WYKORZYSTYWANIA ROZWIĄZANIA QUADRATIC ASSIGNMENT PROBLEM

STRESZCZENIE. **Wstęp:** Kwadratowy Problem Przydziału (QAP) jest jednym z najciekawszych zagadnień optymalizacji kombinatorycznej. Został przedstawiony przez Koopmana i Beckamanna w roku 1957, jako matematyczny model lokalizacji niepodzielnych zadań. Problem ten należy do klasy zagadnień NP.-trudnych. Wymusza to stosowanie do jego rozwiązania metod przybliżonych już dla zadań o niewielkim rozmiarze (powyżej 30).
Mimo że jest ono znacznie trudniejsze niż inne zagadnienia optymalizacji kombinatorycznej, to cieszy się powszechnym zainteresowaniem, ponieważ modeluje ważną klasę problemów decyzyjnych.
Metody: Dyskusji poddano narzędzia sztucznej inteligencji, które pozwoliły rozwiązać problem QAP, między innymi są to: algorytmy genetyczne, Tabu Search, Branch and Bound
Wyniki i wnioski: Sam problem bezpośrednio nie powstał jako model pewnych działań, jednak znalazł on swoje zastosowanie w wielu dziedzinach. Przykładowymi zastosowaniami problemu jest: rozmieszczenie budynków na kampusie uczelnianym, projektowanie rozmieszczenia elementów elektronicznych w układach o wielkiej skali integracji (VLSI), projekt szpitala, rozmieszczenie klawiszy na klawiaturze.

Słowa kluczowe: QAP, problem kwadratowego przydziału, algorytm podziału i ograniczeń, algorytmy genetyczne, symulowane wyżarzanie, algorytm Tabu Search

QUADRATIC ASSIGNMENT PROBLEM (QAP) UND DESSEN ANWENDUNGSLÖSUNGEN

ZUSAMMENFASSUNG. Einleitung: Quadratic Assignment Problem (QAP) ist eine der interessantesten Fragen der kombinatorischen Optimierung. Dies wurde von Koopman und Beckamanna im Jahre 1957 als ein mathematisches Modell des Standortes der unteilbaren Aufgaben vorgestellt. Dieses Problem gehört zur Klasse der NP-schwierigen Fragen. Dies zwingt zur Anwendung einer auf das Näherungsverfahren für Aufgaben mit einer geringeren Größe (über 30) gestützten Lösung.
Obwohl die Lösung viel komplizierter als andere kombinatorische Optimierungslösungen ist, genießt sie ein großes Interesse, weil sie die wichtigste Klasse von Entscheidungsproblemen zu modellieren vermag.
Methoden: Die betreffenden Diskussionen konzentrierten sich auf den Werkzeugen künstlicher Intelligenz, die das QSP-Problem zu lösen erlauben. Dazu gehören unter anderem: genetische Algorithmen, Tabu Search, Branch and Bound.
Ergebnisse und Schlussfolgerungen: Die betreffende Lösung ist nicht als Modell für alle Tätigkeiten anzusehen, es findet jedoch seine Anwendung in vielen Bereichen. Beispiele für die Lösungsanwendungen sind brauchbare Werkzeuge für: eine optimale Anordnung von Gebäuden auf dem Universität-Campus, ein effizientes Entwerfen der Anordnung von elektronischen Komponenten innerhalb der Systeme mit hohem Integrationsgrad (VLSI), eine anwendungsfreundliche Anordnung der Tasten auf der Computer-Tastatur.

Codewörter: Quadratic Assignment Problem, Branch and Bound-Algorithmus, genetische Algorithmen, Simulated Annealing, Tabu Search-Algorithmus.

Izabela Kudelska
Institute of Engineering Management
Poznan University of Technology
Strzelecka 11 St, 60-966 Poznan, Poland
e-mail: izabela.kudelska@put.poznan.pl