Advanced Mathematical Models & Applications
Vol.3, No.3, 2018, pp.179-191

PRACTICAL ASPECTS OF SOLVING COMBINATORIAL OPTIMIZATION PROBLEMS

Urfat Nuriyev[1], **Fidan Nuriyeva**[2,3]

[1]Ege University, Faculty of Science, Department of Mathematics, Izmir, Turkey
[2]Dokuz Eylul University, Faculty of Science, Department of Computer Science, Izmir, Turkey
[3]Institute of Control Systems of ANAS, Baku, Azerbaijan

Abstract. In this paper, problems about increasing the efficiency of the solution methods of Combinatorial Optimization problems are discussed and the method of development of hybrid method concept for Combinatorial Optimization problems is proposed. All problems are discussed in terms of solving them by hybrid method. The essence of the proposed method is as follows: the hybrid method $\{M_H\}$ is created from the given base methods of $\{M_i\}$. The final method consists not only of the sequential application of the base methods, but also the combination of them in a general scheme. Thanks to this method, it has been possible to design efficient algorithms to solve a number of practical problems.

Keywords: combinatorial optimization, system optimization, hybrid methodology, interactive approach, parallelization.
AMS Subject Classification: 90C27, 90C59, 90C90.
Corresponding author: Fidan, Nuriyeva, Dokuz Eylul University, Izmir, Turkey,
e-mail: *nuriyevafidan@gmail.com*
Received: 01 November 2018; Revised: 06 December 2018; Accepted: 18 December 2018; Published: 28 December 2018.

1 Introduction

The large and increasing interest in *Combinatorial Optimization* (CO) problems comes from the fact that many problems arising from practice can be modeled as discrete optimization problems (Gary et al., 2004), (Sergienko & Kaspshitskaya, 1981).

Because of the rapid development of computational networks theory and technology, in recent years many design problems of computational networks and computer systems have been formulated in discrete mathematical programming terms (Cormen et al., 2001), (Gursoy et al., 2017), (Mikhalevich, 1977), (Mikhalevich & Volkovich, 1982), (Nuri et al., 2016), (Sadiq et al., 2000).

Many different solution methods have been developed for the solution of the Combinatorial Optimization problems. The solution experience of optimization problems from this group shows that they are the most difficult problems in terms of computational complexity (time) (Garey & Johnson, 1979), (Romanovsky, 1978). This encourages the development of the theory of discrete extremity problems, more comprehensive learning of the solution methods of these problems, new approaches, methods, algorithms and package programs.

Therefore, intensive research in the field of combinatorial optimization continues. There are the following trends:

- new algorithms are getting proposed;
- rapid development occurs in mathematical plan for preparing analysis of algorithms;

- instead of analyzing algorithms separately, analysis of class of algorithms are made.

The solution algorithms for CO problems are in three main directions:

Firstly, general methods (algorithmic schemas) are prepared to solve large class problems, secondly, efficient algorithms are designed to solve individual class problems (to ensure general schemas too), and finally, special algorithms are prepared for specific practical problems (similar problems).

In the solution of many practical problems, it is necessary to evaluate the efficiency of algorithms according to some parameters. It is necessary to take into account their specific characteristics (precision, time, required memory) and the possibilities for their realization (simplicity of the algorithms calculation scheme, the types of memory required and the frequency of their use, the width of the class to be solved, etc.).

In this paper problems about increasing the efficiency of the solution methods of CO problems are discussed and improved version of concept of hybrid method is proposed.

The proposed method is as follows: the hybrid method $\{M_H\}$ is set from the given base methods of $\{M_\imath\}$. The final method consists not only of the sequential application of the base methods, but also the combination of them in a general scheme.

Also all problems are discussed in terms of solving them by hybrid method. Due to this method, it has been possible to design efficient algorithms to solve a number of practical problems.

2 View on a solution algorithms of combinatorial optimization problems

The solution methods of CO problems are divided into exact and approximate methods (Korbut & Finkel'shtein, 1969). Exact methods guarantee optimal solution; however, the results are not always satisfactory due to the computational complexity (time) and space complexity. In such cases, approximate methods may be used. Although the approximate methods do not guarantee the optimal solution, they can be approached in a reasonable manner and produce solutions in a reasonable time.

Exact Methods: There are two main groups for exact methods (Korbut & Finkel'shtein, 1969), (Kovalev, 1977), (Papadimitriou & Steiglitz, 1998):

Gomory type cutting methods, Benders method, theoretical-group method, and the second group - combinatorial methods.

In combinatorial methods, the following groups are the most common:

- Enumeration
- Dynamic Programming Method
- Sequential Analysis of Variants
- Branch and Bound Method
- The Method of Establishing a Sequence of Solutions.

Dynamic programming has a special place in combinatorial methods (Bellman & Dreyfus, 1962). Dynamic programming is the name given to the analysis methods based on the principle of optimization. This method allows to decrease number of operations but increases the memory usage in comparison to enumeration method. But, when there are a lot of constraints and very large coefficients, dynamic programming is not efficient. These difficulties can be over come by using *the method of the Lagranje multipliers* (Sergienko & Kaspshitskaya, 1981), and *successive approximations method* which is proposed by N.N. Moiseev and by his students (Moiseev et al., 1978).

Lets emphasize that the dynamic programming method for CO problems is a special case of *the method of sequential analysis of more universal and agile variants* proposed by V.S Mikhalevich and N.Z Shor (Mikhalevich, 1977).

The branch and bound method based on eliminating "non-promising variants". Sufficient general scheme of the Branch and Bound method is given in (Finkelshtein, 1976).

Method of successive solutions proposed by V.A. Emelichev (Emelichev & Komlik, 1981) combines Known methods for solution of CO problems. On the basis of this method, the majorant (minorant) evaluation-function decreases in order of decreasing (ascending) order of the solutions of the auxiliary problem for the problems of minimization.

Approximate Methods: Approximate methods do not guarantee an optimal solution, unlike exact methods. However; as it is not possible to perform exact methods for solving large-scale problems, algorithms giving **approximate solutions** are important.

Approximate methods are important according to the following conditions:

a. The exact known methods are not perfect enough because they face difficulties in solving many problems.

b. For some operative problems, the approximate solution in a short computer time is more valuable.

c. The input data of many practical problems is approximate and therefore there is no point in solving them with exact methods.

Therefore, in recent years, *approximate analysis methods of integer problems* are important. A number of approximate methods have been prepared for the solution of current CO problems. A compilation of modern approaches for the approximate solution of CO problems is given in Ausiello et al. (1999), Finkelshtein (1976), Sergienko & Kaspshitskaya (1981), Sergienko et al. (1980).

One of the most commonly used methods is the *heuristic algorithms.* The heuristic algorithm is an algorithm designed according to the nature of the problem and ignores whether the solution is verifiable. However, it usually produces very fast solutions and approaches reasonably to the best possible solution. Considering the worst case, heuristics may show very poor performances; when tough problem example is selected, it may return a result that is too far from the optimum and / or require exponential running time. However, good heuristics may leave behind the performance of the best approximate algorithms in most cases of a problem (Ausiello et al., 1999).

Most of the known heuristics are either "constructive heuristics" that constructs a single appropriate solution recursively, or "enumeration heuristic" which examines a set of appropriate solutions and returns the best result. Constructive heuristics are strongly problem-dependent and often require polynomial time. On the other hand, enumeration heuristics may require exponential running time when the examined set is too large.

Some of the well known heuristics are (Aarts & Lenstra, 1997):

- Greedy algorithms
- Local search
- Genetic algorithms
- Tabu search
- Ant colony etc.

Greedy algorithms: Greedy algorithms, being one of the heuristic methods, is very useful in terms of ease of designing and giving good solutions to optimal solution. Greedy algorithms can provide optimal solutions for some problems. For example, Dijkstra's shortest path algorithm, Chvátal's cluster cover heuristic and the Kruskal algorithm are some of the well-known greedy algorithms (Ausiello et al., 1999), (Cormen et al., 2001).

Generally the characteristics of greedy algorithms can be given as follows:

- Makes one decision at a time
- Uses local knowledge to make decisions
- When making the decision, looks for the most benefit at the that time; so it is called greedy

The greedy method initially sets objects according to some criteria and expands the solution set starting from the empty set. Decides for one object at a time. If the solution at the end is appropriate, the object is added to the current solution; otherwise, they are eliminated. The running time of the greedy algorithm is $O(n)+O(n\ log\, n) = O(n\ log\, n)$ because of the alignment of the grain n number of object and the n number test of conformity.

The quality of the approximate solution depends on the initial sequence. Obviously there is always an optimal sorting for each problem; however, it is unlikely to find such sorting in the polynomial time for all instances of a problem that is difficult to compute. In some cases, however, there may be simple sortings in which the greedy method provides good approximate solutions (Ausiello et al., 1999).

3 Practical aspects of combinatorial optimization problems

Exact solution methods were generally prepared in the early stages of the development of CO methods (Finkelshtein, 1976), (Korbut & Finkel'shtein, 1969), (Sergienko et al., 1980). However, theoretical evaluations and calculation experiments show that most of the CO problems are in the *class of NP* (Finkelshtein, 1976), (Garey & Johnson, 1979), (Gens & Levner, 1979), (Sergienko & Kaspshitskaya, 1981), (Sergienko et al., 1980). In order to solve these problems, the calculation time of all known exact algorithms increases exponentially according to the size of the problem and it seems that there are no exact solution algorithms with increasing polynomial time. Therefore, approximate methods have been developed to find solutions that are close to the optimal solution (Finkelshtein, 1976), (Sergienko & Kaspshitskaya, 1981), (Sergienko et al., 1980). Thats why, taking into account the theoretical results, practical aspects are more important. These theoretical results show the *worst aspects* of these algorithms. But, for their calculations, their *average results* are more important. It is important to assess its practical complexity in the preparation of each algorithm.

Lets consider the fundamental characteristics of the process of solving optimization problems in computer systems. Experience in the preparation and implementation of optimization problems in computer systems shows that these features are important in selecting the solution methods of these problems (Gary et al., 2004).

1. Real planning-economic problems that can be modeled as integer programming problem are large-scale. Certain known exact methods for solving Integer Programming (IP) problems are not efficient because they require huge computational time and computer memory.
2. Another characteristic feature of technical economic problems is that their input data are not certain and that when a production plan is in line with another production plan or any plan is carried out, it changes.
3. When the properties of the optimal solutions are examined, it can be seen that there are multiple optimal points or point close to optimal. If there is a rule to distinguish these points, it must be added to the model.

4. In many cases when the input data changes, a series of solutions that are close to the optimum is more appropriate than solution that is far from the optimum.

Thus, when analyzing the plans within these solutions (for example, to accommodate plans of other production complexes) one of them is chosen. This is also important for selecting the final solution by considering the non-formal criteria by the expert. For each algorithm that has been experimented by analyzing the solution experience of practical problems, it is necessary to examine the followings:

1. Does the calculation of the value of the objective function in comparison to other algorithms require less operation?

2. Does the algorithm require less computational time?

3. Is there less computer memory needed to implement the algorithm?

4. Is the algorithm convergent; Does it give information about the calculation process?

5. How is the sensitivity of the algorithm to the calculation errors;

6. How much precision of the solution is satisfied;

7. What is the relation with the accuracy of the solution get from this algorithm and a running time of the algorithm;

It is necessary to solve the CO problems and underline some situations related to their practical aspects and to take appropriate measures.

Lets underline that difficulties met in solving different CO problems are independent from the used methods and are compatible with the features of these problems.

Although heuristic methods provide a very different solution than the optimum solution of these problems, there is no other method which can find the optimal solution with the optimal or integer solution at the desired time. Secondly, even when these problems are successfully solved with known CO methods, it should be approached suspectively that the solution is optimal.

System Approach: The actual processes desired to be modeled like Integer Programming (IP) problems are still more complex without considering the accuracy of the input data.

In these cases, the solution of IP problems may not be adequate. Thus, the efficiency of the solution process in optimization problems is determined not by the speed and precision of the calculation procedure, but by the organization of the whole process. In this case, the *system approach* proposed by V.M. Glushkov (Glushkov, 1980) is necessary, which involves the process of solution of the given problem, as well as the regulation of the mathematical model. The fundamental parameters of the mathematical model of the optimization problems can be the constraints of the problem and the criterion of optimization.

This approach takes into account the hierarchy of optimization models. Thus, the final decision in both the accepting of the optimal solution and the emphasis on the different criteria, and the determination of the limits of the appropriate set of solutions, is made by the active participation of the experts. This time, the classical optimization problems that we know come up as local sub-problems in different stages of optimization.

One of the important features is the interconnection of different levels of models, with the multi-criteria of system optimization and changing the appropriate set of solutions. A similar approach was suggested in study Gary et al. (2004).

When the problems of planning-economics are solved, there will be a connection between the managers working at different levels and the problems of the project construction engineers working on different parts of the project.

Interactive approach: According to the general principle of the system approach, optimization problems and levels are determined according to the obtained software. Therefore, many methods and tools are used at the time of practical optimization of planning and management problems, among which the *interactive approach* is more important.

Methods and approaches have been developed to improve the efficiency of these systems during the preparation of optimization systems. One of these is the *manageable optimization*, so that the programmer can contribute to the optimization process, i.e., the programmer can make the change in the interactive regime with the system. Lets underline that in this case the *interactive regime* is a very efficient tool. In particular, when it is necessary to examine the mathematical model of the problem, the calculation process is very efficient when it monitors the effect of different parameters on the optimization process.

The solution of complex optimization problems in the interactive regime does not only reduce computational time, but also it is a new efficient approach to finding this solution (Mikhalevich & Volkovich, 1982).

Let's underline that the optimization methods put into such systems must have the following characteristics: they must first be able to work in "to continue from where it stopped", that is, if a part of the solution has been found, it should be able to find the exact solution by continuing, where the input data and the conditions of the problem changes, it must fix the solution.

The interactive system allows:

1. It takes into account non-formalized conditions which are not considered in the economic-mathematical model of production

2. Makes analysis of the solutions offered by the employees of the plan and get by computers

3. Makes the computational experiments

4. Operate the economic analysis of the taken variants of production programs

5. Covers all participants involved in the preparation, finding and acceptance of the solution

6. Ensures that the planning process is continuous

7. Prepares the employees of the planning branches of the organization to use the new plan design technologies.

4 Hybrid methodology of solving discrete optimization problems

Starting from what is mentioned in the previous paragraphs, a very important problem follows: Is it possible to prepare such a method that is less dependent on the size, model and structure of the variables for solving large class problems. Of course it is not possible to provide such a universal method that fully satisfies the conditions given above. However, we can get some results by considering the parameters of the learned process in a complex way and by using the flexibility of the method.

The concept of the core: It is known in discrete mathematics that by discarding some subsets of zero dimension from the whole set many features of the set are exposed. Similar observations exist in CO problems. Such a rule is known for large-size IP problems with few constraints (Buzytskiy & Freyman, 1980). This allows to separate the *core* of the problem. The *core concept* is used to give a set of variables that can be changed in a set of optimal or near-optimal solutions. It is known that the size of this type of *core* is very small according to the size of the given problem (Buzytskiy & Freyman, 1980).

Obviously, if the size of the *core* is not large, we can get the final solution of the problem with known directed selection methods.

Therefore, it is possible to solve the size problem with a *complex (system)* approach:

1. Variables divided into two *significant* and *non-significant* classes (choosing the core of the problem).

2. Selecting the core with fast methods (heuristic methods provide this condition)
By using such methods we can get a good approximate solution. A good solution allows you to better determine the core set. The smaller the core, the better the solution.

3. It is necessary to use such a method that only the input data of the problem should be used at every step when it seeks the optimal solution. It should be appropriate in both parallel calculation and interactive regime. Thus, there is the possibility of using different methods in this solution process and there is no need for additional memory to place intermediate results.

Hybrid method: The so-called *hybrid method* is very promising. These methods are based on the complex use of different approaches.

Successful solution of individual large-scale CO problems by different methods shows that the possibilities of these methods are wide. In order to solve CO problems, different algorithms should be combined in a scheme to get a fast and efficient algorithm (Romanovsky, 1978).

The idea of getting a better algorithm with simple approximate algorithms series is not new. Good results in this area has been introduced by Yu.I. Zhuravlev (Zhuravlev, 1978), (Zhuravlev, 1966) to produce highly efficient recognition algorithms from a series of simple recognition procedures.

In the study Adamenko (1982), the concept of *safety of the algorithm* is defined. This concept informs the *ratio* of the size of the set with the given algorithm to the size of the set where the extremal problem is defined. It has been shown that the composition of the algorithms has higher safety than their components.

Thus, the success of the solution of the CO problems is based on the complex implementation of different methods by combining different methods in a flexible manner.

One of these approaches is the combination of heuristic and regular optimization methods.

All problems in the article were examined in terms of the application of the hybrid method.

The hybrid methods discussed in the paper are consist of two stages. First, heuristically self-organized, quick but rough heuristic algorithms are used to find a good starting solution. Considering the structure of the problem and the numerical parameters, the variables expressed by numerical values are determined. This allows it to organize the iterations according to the target and to find a good approximate solution. According to its importance, the division of variables into two significant and non-significant classes allows solving the size problem in solving large-scale problems. The found suboptimal solution "resuming" to the complete solution by regular methods.

Hybrid Methodology: Consider the general *methodology* of the hybrid method. The essence of the proposed method is as follows: the *combined* (hybrid) method M_H is established by the given $M_\imath$ base methods.

The final method is not only by the sequential application of the base methods, but by combining them into a general schema. Obviously, this scheme will not be the same for all problems, there will be different modifications according to the characteristics of the problem.

To illustrate how the schema is written above, consider how the two most widely published methods - heuristic and branch and bound methods are combined in this way.

Heuristic methods allow solving large class problems. These methods take into account the many characteristics of the problems and can adapt to each problem depending on the situation encountered. In spite of these advantages of heuristic methods, in many cases their efficiency is very low. Such a solution may be very different from the optimal solution. On the other hand, a disadvantage of the heuristic algorithm is that they are *irregular*, so that it is not possible to evaluate how different the solution is from the optimal solution.

However, in many cases it is necessary to know this difference and as long as the optimal solution allows time, it is desirable to get closer. In this respect, it is very promising to improve the solution found with heuristic methods with the branch and bound scheme.

As it is known, the *Branch and Bound method* is a directional selection procedure by cutting out "non-promising" variants. The advantage of the Branch and Bound method is that it is agile, considering the characteristics of the given CO problem and using these and other efficient methods in the scheme.

The general scheme of these methods allows not only to establish approximate solution sequences, but also to assess how different the founded solution of objective function is from the optimal solution. Furthermore, the Branch and Bound method is less sensitive to rounding errors.

To establish an efficient hybrid method on the basis of heuristic methods M_H and Branch-Bound methods M_{BB}, these methods must satisfy the following conditions:

- M_H and M_{BB} must be data-dependent, that is, the result of one of them should be given to the other without any change
- the general calculation process should be iterative; there should be at least approximate solutions
- the algorithm should include a variable that provides information about the state of the calculation process and can be used to manage it

To provide these requirements, M_H must be constructed as follows:

a. M_H be able to be used in the initial stage,

b. M_{BB} solution tree should be set according to the approximate solution,

c. it should be possible to set a relaxation problem on each top of the tree.

The condition (a) allows one to find a solution to the problem given in the initial stage (the good solution now allows to discard many non-promising variants in the beginning). Therefore, we can finish the calculations at any convenient time and use the best solution found until then.

The conditions (a) and (c) allows to evaluate cloosenest of the solution to the optimal solution.

For example, lets consider the general Boolean Programming problem:

$$F_0\{x_1, x_2, \ldots, x_j, \ldots, x_n\} \to max \tag{1}$$

under constraints

$$F_i\{x_1, x_2, \ldots, x_j, \ldots, x_n\} \leq T_i, i = 1, 2, \ldots, m, \tag{2}$$

$$x_j = 0 \vee 1, i = 1, 2, \ldots, n. \tag{3}$$

Here, F_i $(i = 0, 1, 2, \ldots, m)$ is any objective function, T_i - is constant. The heuristic part of the algorithm is based on the choice of the superiority function. Most of the heuristic methods are based on the following assumption. A number that is obtained by a heuristic method is assigned to each variable. Selection of variables is determined by this number. The variable that corresponds to the greater number is chosen. Formally, the superiority of one of the variables is expressed as the probability evaluation of the value of this or other variable as 1. However, instead of the numerical evaluation of the choice, it can only be discussed about their mutual comparison, that proportioning in the cluster formed by them. In other words, although it is not possible to evaluate that each variable can be chosen, it is possible for each variable pair to give a numerical value for one of these variables to be more useful or preferable than the other. According to this model, preference is considered as a set of measurable factors, and the selection of variables is expressed as a process of creating a set (Gursoy & Nuriyev, 2016).

In each step of this process, a variable is selected proportionally to its preference.

$$x_{i1} \succ x_{i2} \succ \ldots \succ x_{in}.$$

Lets select the first variable p (as long as (2) is provided) and lets find heuristic solution:

$$X_H = (\underbrace{1, 1, ..., 1}_{p}, \underbrace{0, 0, ..., 0}_{n\text{-}p}).$$

Thus, there is no need for the addition of the vector $\overrightarrow{X_H}$ and the additional procedure for the classification of the variables in an important and unimportant way. They are determined at the time of the heuristic method. Thus, the heuristic method is used to classify the variables as important and non important.

Lets introduce one dimensional problem (4) - (6) with the same objective function of the problem (1)-(3):

$$F_0\{x_1, x_2, \ldots, x_j, \ldots, x_n\} \rightarrow max, \tag{4}$$

$$F_i\{x_1, x_2, \ldots, x_j, \ldots, x_n\} \leq T_i, i = 1, 2, \ldots, m, \tag{5}$$

$$0 \leq x_j \leq 1, j = 1, 2, \ldots, n. \tag{6}$$

Here, $F_0\{x_1, x_2, \ldots, x_j, \ldots, x_n\}$ (2) the change (artificial) condition of the system of constraints. For the first p and the last $(n-p)$ variables, we can construct the individual trees of variants (P and W trees) in such a way that when moving on the tree, value of the object function of the relaxation problem does not increase and the dominant sub-trees are easily detected.

Therefore, when the value of the objective function of any relaxation problem is smaller than the record, the sub-trees with root A and the sub-trees it dominates are discarded. The following problem is solved on each top of this tree (search on tree W).

$$F_0\{\overline{x_1}, \ldots, \overline{x_p}, x_{p+1} \ldots, x_n\} \rightarrow max, \tag{7}$$

$$F_i\{\overline{x_1}, \ldots, \overline{x_p}, x_{p+1} \ldots, x_n\} \leq T_i^A, i = 1, 2, \ldots, m, \tag{8}$$

$$x_j = 0 \vee 1, i = 1, 2, \ldots, n. \tag{9}$$

Here, $\overline{x_1}, \ldots, \overline{x_p}$ variables has certain values. The number of variables of the (7) - (9) problems is less $\forall i, T_i^A < T_i$ and therefore more simple than the problems (1) - (3). Let's underline the following. Here, it is important to know the degree to which everything is superior to the value of the function of the variables. Therefore, you need to choose the best suboptimal solution by changing the function of superiority.

Such a question arises: it would not be possible to find a solution near the X^* optimal solution as the starting point X_0 (it helps in finding this point in the heuristic method).

In the proposed method, the starting point X_0 is not given, it must be searched and this searching is made in such a way that X_0 point is dependent on X^*.

The essence of the proposed method is the finding suboptimal X_0 point with the help of the priority function. The control of the optimal solution begins with the suboptimal solution X_0.

According to the hybrid method, the input parameters of the problem are based on system analysis, which allows to determine the variables according to their priority.

The degree of priority of variables is expressed numerically. Here, the search procedure allows to organize according to the priority of the components of the vector $\overrightarrow{X_H}$.

Therefore, in any iteration, the calculation process is finished and the solution is considered as the approximate solution.

Thus, the differentiated feature of the proposed method is that it is a iteration type, that it finds a good solution at the beginning of the calculation process, and that it finds a solution that is closer to the optimal solution than the reserved resources (time, computer memory, etc.).

In all combinatorial procedures, three operations are performed in each iteration: branching, calculating the objective function and comparing it with the best solution. All heuristic methods use the priority function according to the same scheme to find the appropriate solution with sequential operations. Therefore, the structure of these algorithms is easily disintegrated into separate procedures. This allows the module principle to be implemented in designing the programs.

Parallelization: Different problems in different application package programs (UPP) are performed as separate modules. In general, these methods do not adapt to the conditions of separate problems. That is, they are less flexible. Their structure is hard and incompatible. As a result, these methods provide different quality solutions for different input methods.

In order to solve such problems, the parameters of the calculation procedures used in the solution process can be arranged and the approaches that have minimized the use of resources such as calculation time by taking into account the characteristics of the solved problem are promising.

It is advisable to design a set of procedures so that each module has a functional durability, i.e., only one function should be performed and the program's management parameters. Each program is obtained by successive use of a number of modules. For example, each new heuristic method is obtained only by changing subprogram calculation of the priority function. By changing the priority function, we choose the best record and the best evaluation problem.

The best record and best evaluation function reduces the uncertainty in the evaluation of the objective function and hence the efficiency of the selection algorithm increases.

The formation of multiprocessor computers provides a good basis for the realization of system optimization methods.

The solution of problems in multiprocessor computer systems in principle requires the preparation of new kinds of algorithms-parallel algorithms. The properties of combinatoric problems easily allow for the parallelization of combinatorial methods to solve them (Ferreira & Pardalos (1996)).

One of the method of creating parallel algorithms is the replacement of sequential algorithms in modern single-processor computers.

In the proposed hybrid method, the parallel calculation is transformed into a "necessity" from a desire. Because, in this method, the search tree is built in such a way that each branch has its own problem and consecutive calculation time these problems are in line. Thus, the value of each hill is compared to the same record for all peaks. If a record is found in any branch better than the previous one, then the old record is replaced with a new one, which changes the search line in other branches. Thus, the parallelization of the calculations yields in calculation time by not only k times (where k - is the number of processors), but more by eliminating many variations due to record value.

5 Conclusion

Thus, the different feature of the proposed method is its iterative type. Another different feature of the method is that it finds the best possible solution according to the given calculation sources by finding a good suitable solution at the beginning of the calculation process. I In other words, the larger the provided resources (time, machine power, etc.), the closer the solution will be to the optimal.

The authors applied the proposed general method to solve different problems in different years: In Berberler & Nuriyev (2010) and Nikitin & Nuriev (1983), a hybrid of the Greedy and Dynamic programming methods, is applied to one-dimensional knapsack and cutting problems. In the study Nuriev (1983) and Nuriyev & Dundar (2002) for the solution of multidimensional knapsack problem a hybrid of greedy, the method of sequential analysis of variants, branch and bound technique, method of successive solutions were proposed. In studies Kizilates & Nuriyeva (2013b), Kızılateş & Nuriyeva (2013c) parameter have been added to the hybrid methods for solution Traveling Salesman Problem. In the studies Atilgan & Nuriyev (2012); Nuriyev et al. (2018), an iterative method have been applied by using hybrid of the greedy algorithms. In the studies Guler et al. (2012) and Kizilates & Nuriyeva (2013a), only hybrid of greedy algorithms was used.

Experimental and theoretical results show that the proposed method is efficient.

References

Aarts, E. & Lenstra, J.K. (1997). *Local Search in Combinatorial Optimization*. John Wiley & Sons, London, 512.

Adamenko, G.M. (1982). Reliability of optimization algorithms and some ways to increase it. *Doklady Akademii Nauk BSSR*, Minsk, *26*(5), 418-420 (in Russian).

Atilgan, C. & Nuriyev, U.G. (2012). *Proceeding of the fourth International Conference Problems of Cybernetics and Informatics. Section VII, Control and optimization 4*, 111-114.

Ausiello, G., Cerscenzi P., Kann V., Marchetti-Spaccamela, A. & Protasi, M. (1999). *Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties*, Springer, Berlin, 524p.

Belenkiy, V.Z., Volkonskiy, V.A., Ivankov, S.A., Pomanskiy, A.B. & Shapiro, A.D. (1962). *Iterative Methods in Games Theory and Programming*, Moscow, Nauka, 239p. (in Russian).

Bellman, R.E. & Dreyfus, S.E. (1962). *Applied Dynamic Programming*, Princeton University Press, New Jersey, 364p.

Berberler, M.E. & Nuriyev U.G. (2010). A new heuristic algorithm for the one-dimensional cutting stock problem *Applied and Computational Mathematics. 4*(1), 19-30.

Buzytskiy, P.L. & Freyman, G.A. (1980). The application of analytical methods in combinatorial problems. *Izvestiya Akademii Nauk SSSR, Teknicheskaya Kibernetika. 2*, 12-18 (in Russian).

Cormen, T.H., Leiserson, C.E., Rivest, R.L. & Stein, C. (2001). Introduction To Algorithms. *The MIT Press*, Cambridge, 1203p.

Emelichev, V.A. & Komlik, V.I. (1981). A Method for Constructing a Sequence of Plans for Solving Discrete Optimal Problems. Moscow, Nauka, 208p (in Russian).

Ferreira, A. & Pardalos, Panos (Eds.). (1996). Solving Combinatorial Optimization Problems in Parallel Methods and Techniques *Methods and Techniques*. Springer, 274p.

Finkel'shtein, Yu.Yu. (1976). Approximate Methods and Applied Problems of Discrete Programming Moscow, Nauka, 264p. (in Russian).

Garey, M.R. & Johnson, D.S. (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, San Francisco, 340.

Gary, A. Kochenberger, F.G., Bahram, A. & Cesar, R. (2004). A unified modeling and solution framework for combinatorial optimization problems. *Operations Research-Spektrum*, *26*(2), 237-250.

Gens, G.V. & Levner, Y.V. (2004). Discrete optimization problems and efficient approximate algorithms. *Izvestiya Akademii Nauk SSSR, Teknicheskaya Kibernetika*, *6*, 9-20 (in Russian).

Glushkov, V. M. (1980). About system optimization. *Cybernetica*, 89-90.

Güler, A., Nuriyev, U.G., Berberler, M.E. & Nuriyeva, F.U. (2012). Algorithms with Guarantee Value for Knapsack Problems. *A Journal of Mathematical Programming and Operations Research*, *61*(4), 477-488.

Gursoy, A., Tekin, A., Keserlioğlu, S., Kutucu, H., Kurt, M. & Nuriyev, U. (2017). An improved binary integer programming model of the Band Collocation problem. *Journal of Modern Technology and Engineering*, *2* (1), 34-42.

Gürsoy, K. N. & Nuriyev, U. (2016). Some inequalities for algebra of fractions and its applications. *Advanced Mathematical Models and Applications*, *1* (1), 1-13.

Kızılateş, G. & Nuriyeva, F.U. (2013a). A New Hybrid Heuristic Algorithm for Solving TSP. *Anadolu University Journal of Science and Technology – B Theoretical Science*, *2* (2), 143-148.

Kızılateş, G. & Nuriyeva, F. (2013b). A Parametric Hybrid Method for the Traveling Salesman Problem. *Mathematical and Computational Applications*, *18* (3), 459-466.

Kızılateş, G. & Nuriyeva, F. (2013c). A two-parameter hyperheuristic method for the Traveling Salesman Problem. *The Global Reach of Industrial Engineering*, Istanbul, Turkey, June 26-28, 242.

Korbut, A.A. & Finkel'shtein, Yu.Yu. (1969). *Discrete Programming*. Moscow, Nauka, 368p.

Kovalev, M.M. (1977). D*iscrete Optimization (Integer Programming). Izdat.: Belorus Univ.*, 192.

Mikhalevich, V.S. (eds). (1977). *Computational Methods for Selection of Optimal Design Works*. Kiev, Naukova Dumka, 132p.

Mikhalevich, V.S. & Volkovich, V.L. (1982). *Computational Methods for the Analysis and Design of Complex Systems*. Moskow, Nauka, 286 (in Russian).

Moiseev, N.N., Ivanilov, Yu.Yu. & Stolyarova, E.M. (1978). *Optimization Methods*. Moskow, Nauka, 352p. (in Russian).

Nikitin, A.I., & Nuriev, U.G. (1983). On a method of the solution of the Knapsack Problem. *Kibernetika*, *2*, 108-110 (in Russian).

Nuri, E., Nuriyeva, F. & Nasiboğlu, E. (2016). A fuzzy model of the problem of optimal placement and integration of offshore oil and gas platforms. *Journal of Modern Technology and Engineering*, *1*(1), 30-38.

Nuriyeva, F. & Kızılateş G. (2013). A parametric algorithm for TSP based on greedy heuristic. *Shevchenkivska Vesna, XI interdisciplinary conference of students, postgraduates and young scientists*, Kiev, Ukrain, March 18-22, 32-36.

Nuriev, U.G. (1983). A hybrid method for solving the multidimensional Knapsack Problem, [Preprint]. 83-43. *Akad. Nauk Ukrain. SSR, Inst.*, Inst. Kibernet., Kiev, 28p (in Russian).

Nuriyev, U. & Dundar, P. (2002). A Self-organizing Algorithm for solving of a multidimensional Knapsack Problem. *Yoneylem Araştırması Dergisi (Transactions on Operational Research)*, *(13, 14)*(1,2). 13-28.

Nuriyev, U. G., Uğurlu, O. & Nuriyeva, F. (2018). Self Organizing Iterative Algorithm for Travelling Salesman Problem. *18th IFAC Conference on Technology, Culture and International Stability (TECIS-2018)*, ELSEVIER, ScienceDirect IFAC Papers Online 51-30, 268-270.

Papadimitriou, C. H. & Steiglitz, K. (1998). *Combinatorial Optimization: Algorithms and Complexity*, *Dover Publications Inc.*, New York, 512p.

Romanovsky, I.V. (1978). *Algorithms for solving of extreme problems,* Moskow: Nauka, 240 (in Russian).

Sadiq, M. Sait & Habib, Y., (2000). *Iterative computer algorithms with applications in engineering: solving combinatorial optimization problems.* Wiley IEEE Computer Sosiety Pr., 410p.

Sergienko, I.V. & Kaspshitskaya, N.F. (1981). *Models and Methods of Computer Solutions of Combinatorial Optimization Problems.* Kyiv, Naukova Dumka, 288p. (in Russian).

Sergienko, I.V. Lebedeva, T.T. & Roshchin, V.A. (1980). *Approximate Solution Methods for Discrete Optimization Problems.* Kyiv, Naukova Dumka, 273p. (in Russian).

Zhuravlev, Yu. I. (1978). Correct algebras over sets of incorrect (heuristic) algorithms, I, II, III. *Kibernetika*, 4, 14-21, 6, 21-27, 2, 35-43 (in Russian).

Zhuravlev, Yu. I. (1965). Local algorithms for the calculation of information, I, II,. *Kibernetika*, 1, 12-19, 2, 1-11 (in Russian).