A new model for equipment selection and transfer line design problem

Pavel Borisovsky ∗ Alexandre Dolgui ∗∗ Sergey Kovalev ∗∗

∗ Omsk F.M. Dostoevsky State University, 55a, pr. Mira, 644077
Omsk, Russia (e-mail: borisovski@mail.ru).

∗∗ Scientific Methods for Industrial Management Department (MSGI),
Industrial Engineering and Computer Science Division (G21), Ecole
des Mines de Saint Etienne, 158, cours Fauriel, 42023 Saint Etienne
Cedex 2, France (e-mail: dolgui@emse.fr, kovalev@emse.fr).

Abstract: We suggest a new mathematical model for the equipment selection and the design of a transfer line intended for a mass production of machine parts of the same type. The transfer line is a sequence of (work)stations equipped with processing modules (blocks) each of which performs specific operations. Each machine part moves along the transfer line in the same direction and undergoes a given set of operations at the stations. There is the same cost associated with each station and different costs associated with the blocks. The problem is to determine the number of stations, to select a set of blocks from the set of available blocks and to assign these blocks to the stations so that each operation is performed exactly once and the total cost is minimized. The peculiarity of the problem is that all operations of the same station are performed in parallel, the assignment of the blocks and operations to a station are restricted by exclusion and inclusion relations, and the processing order of the operations on the transfer line must comply with precedence relations. We suggest a reduction of this problem to a set partitioning type problem. The reduction is based on a new concept of a locally feasible station.

Keywords: Operations research; Production systems; Optimization problems; Mathematical models; Preprocessing.

1. INTRODUCTION

The following problem is addressed. A transfer line has to be designed for an execution of a given set \( N = \{1, \ldots, n\} \) of operations for a mass production of machine parts of the same type. By a transfer line we call a sequence of (work)stations such that a subset of the required operations is executed on the first station, then another subset of the operations on the second station, and so on until each operation of the set \( N \) has been executed exactly once. Stations have to be equipped with a number of processing modules, which we call blocks. Each block is associated with a set of operations it can perform. Operations of all blocks assigned to the same station are performed in parallel so that the processing time of the machine part on the station is determined by the longest operation of this station, and the transfer line cycle time is determined by the longest operation of all \( n \) operations. If an upper bound on the transfer line cycle time is given, then we assume without loss of generality that the duration of the longest operation in \( N \) does not exceed this upper bound.

The set of available blocks, \( B \), is assumed to be given. Plural exclusion relations are imposed on this set. They are represented by a collection \( E \) of subsets \( E' \subset B \) such that all blocks of \( E' \), \( E' \in E \), cannot be assigned to the same station but any proper subset of \( E' \) can be assigned to the same station. These relations serve for resolving situations where some tools cannot be simultaneously activated on the same station due to the conflict of their physical characteristics, for example, dimensions. We assume without loss of generality that there are no two sets in \( E \) containing one another.

Plural inclusion and binary precedence relations are given on the set \( N \) of operations. Inclusion relations are represented by a collection \( I \) of subsets \( I' \subset N \) such that all operations of \( I' \), \( I' \in I \), must be assigned to the same station. These relations model situations in which the required precision of some operations can be lost if the machine part moves between these operations. We assume without loss of generality that all the sets in \( I \) are non-intersecting.

If operation \( i \) precedes operation \( j \), which is denoted as \( i \rightarrow j \), then \( j \) cannot be executed on the station containing \( i \) or any preceding station. Precedence relations are transitive and irreflexive. We assume that they are represented by an acyclic directed graph \( G_o = (N, A_o) \), in which there is an arc \( (i, j) \in A_o \) if and only if operation \( i \) precedes operation \( j \).

A cost \( q(b) > 0 \) is associated with each block \( b \in B \) and a cost \( C > 0 \) is associated with each station. Let \( m_0 \) and \( n_0 \) denote given upper bounds on the number of stations and the number of blocks of the same station, respectively. The problem is to determine the number of stations, \( m \), the sets of blocks assigned to these stations, \( W_1, \ldots, W_m \), and the sequence of the stations such that \( m \leq m_0, |W_k| \leq n_0, \)
By Dolgui et al. [2004] and Belmokhtar et al. [2006], each operation of the set \( N \) is executed exactly once, exclusion, inclusion and precedence relations are satisfied, and the total cost,
\[
mC + \sum_{k=1}^{m} \sum_{b \in W_k} q(b),
\]
is minimized. In the sequel, we will also use notation \( W_r \) for a station.

We denote this problem as problem P. It was first formulated by Dolgui et al. [2004] and Belmokhtar et al. [2006], who emphasized its practical importance and outlined differences from the line design problems (for the latter problems, see reviews by Baybars [1986], Erel and Sarin [1998], Scholl [1999] and Rekik et al. [2002]) and problems of the design of transfer lines with sequential operations (for these problems, see papers by Graves and Lamar [1983], Graves and Holmes Redfield [1988], Bukchin and Tzur [2000], Bukchin and Rubinovitz [2002], Delorme et al. [2006b], and Dolgui et al. [2006a]). Problems similar to P have been observed in Belarusian and French factories MZAL and PCI-SCEMM, see Guschinskaya [2007] and Delorme et al. [2009], respectively.

Dolgui et al. [2004] and Belmokhtar et al. [2006] suggested two integer programming formulations for problem P and described extensive computer experiment with these formulations by using ILOG CPLEX software. Dolgui et al. [2006c] presented a reduction of problem P to a shortest path problem. Dolgui et al. [2008], Guschinskaya and Dolgui [2009], and Dolgui and Ilnatsenka [2009a,b] studied a problem differing from problem P in that operations of different blocks are performed sequentially or in a given order. One of the observations made in Belmokhtar et al. [2006] about the exact solution of problem P is that the computational effort drastically depends on the number \( n \) of operations. In this paper, we present a combinatorial model and a solution method which can be useful in overcoming this difficulty, especially in the case, where the number of blocks assigned to the same station in an optimal solution is sufficiently small. Recall that this number is upper bounded by \( n_0 \). In real-life applications, \( n_0 \in \{2,3\} \) is often satisfied because at most two modules are usually assigned horizontally at both sides of the transfer line and at most one block is usually assigned over it at the same station.

We suggest a reduction of problem P to a set partitioning type problem. Section 2 presents our reduction and corresponding mathematical programming model. A key element of the reduction is a new concept of a locally feasible station. Ideas of a preprocessing procedure are given in Section 3. The results of computer experiments are given in Section 4. The paper concludes with a summary of the results.

2. REDUCING PROBLEM P TO A SET PARTITIONING TYPE PROBLEM

It is convenient to introduce the following terminology. We call a collection of blocks \( W \subseteq B \) a locally feasible station (LF-station) if \( |W| \leq n_0 \), each operation of \( W \) belongs to exactly one block of \( W \) and the exclusion, inclusion and precedence relations are satisfied for the blocks and the operations of \( W \). Regarding the inclusion relations, it means that either no operation of a set \( I' \subseteq I \) belongs to a block of \( W \) or all operations of such a set belong to the blocks of \( W \). Regarding the precedence relations, no two operations \( i \) and \( j \) such that \( i \rightarrow j \) can be assigned to the same LF-station \( W \). As for the exclusion relations, an LF-station \( W \) should not contain all the blocks of any set \( E' \subseteq E \).

We call a sequence \( L = (W_1, \ldots, W_m) \) of LF-stations such that the machine part moves along them in the indicated order a feasible transfer line, if \( m \leq n_0 \), each operation of the original set \( N \) belongs to exactly one block and the precedence relations are satisfied for all the \( n \) operations. Notice that the inclusion and exclusion relations are necessarily satisfied for a feasible transfer line because it consists of the LF-stations.

Let \( W \) and \( L \) denote the set of all LF-stations and the set of all feasible transfer lines, respectively. It is easy to see that an optimal solution to problem P is a feasible transfer line \( L \subseteq \mathcal{W} \) with the minimum total cost defined by (1).

Set \( \mathcal{W} \) of LF-stations can be constructed as follows. Let blocks of the set \( B \) be numbered \( B_{1,\ldots,K} \), where \( T = |B_c| \). We can identify each LF-station \( W \subseteq W \) with a \( k \)-tuple \((z_1, \ldots, z_k)\) of indices of its blocks or with a 0-1 vector \((y_1, \ldots, y_T)\) such that
\[
y_i = \begin{cases} 1, & \text{if } B_i \in W, \\ 0, & \text{if } B_i \notin W. \end{cases}
\]
Clearly, \( W \) is a subset of the set of all the \( k \)-tuples \((z_1, \ldots, z_k)\) such that \( 1 \leq k \leq n_0 \) and \( z_i, i = 1, \ldots, n_0 \), are all the distinct numbers of the set \( \{1, \ldots, T\} \), or alternatively, it is a subset of all the \( T \)-dimensional 0-1 vectors. Therefore, \(|W| \leq \sum_{k=1}^{n_0} |T| = \frac{T(n_0 + 1)}{2} = O(T^{n_0})\) and \(|W| \leq O(2^T)\), i.e., \(|W| \leq O(\min(T^{n_0}, 2^T))\). The set \( \mathcal{W} \) can be constructed in \( O(Poly(n, T, |E|) \min(T^{n_0}, 2^T)) \) time by considering every \( k \)-tuple or every 0-1 vector mentioned above and checking the corresponding station for local feasibility. Here \( Poly(n, T, |E|) \) is a polynomial of \( n, T \) and \( |E| \). This polynomial depends on \(|I|\) as well, but as soon as the sets of operations are non-intersecting and, therefore, \(|I| \leq n\), we do not include \(|I|\) in the arguments list.

A feasible transfer line \( L \subseteq \mathcal{W} \) with the minimum total cost can be constructed as follows.

Set \( K = |\mathcal{W}| \). Enumerate LF-stations \( W_1, \ldots, W_K \). Calculate cost, \( c_i \), of each LF-station \( W_i \):
\[
c_i = C + \sum_{b \in W_i} q(b), \quad i = 1, \ldots, K.
\]
Introduce 0-1 variables \( x_i, i = 1, \ldots, K \):
\[
x_i = \begin{cases} 1, & \text{if LF-station } W_i \text{ is included in a transfer line,} \\ 0, & \text{otherwise.} \end{cases}
\]
Thus, every 0-1 vector \( x = (x_1, \ldots, x_K) \) corresponds to an unordered transfer line \( UTL(x) = \{W_{i_1}, \ldots, W_{i_m}\} \) such that \( \{r_1, \ldots, r_m\} = \{i \mid x_i = 1, i = 1, \ldots, K\} \).

Identify set \( S_j \) of LF-stations which contain operation \( j \):
\[
S_j = \{W_i \mid j \in W_i, i = 1, \ldots, K\}
\]
for \( j = 1, \ldots, n \).

Let \( \text{Oper}(W_i) \) denote the set of operations of LF-station \( W_i \). Introduce directed graph \( G_w = (W, A_w) \), in which the set of LF-stations is the set of vertices, and there is
an arc \((W_p, W_r) \in A_w\) if and only if \(W_p\) and \(W_r\) have no common operation and there are operations \(i \in \text{Oper}(W_p)\) and \(j \in \text{Oper}(W_r)\) such that \(i \rightarrow j\). Let \(Y_1, \ldots, Y_a\) be all cycles in \(G_w\). Consider the following set partitioning type problem, which we denote MinSPP.

Problem MinSPP:

\[
\min \sum_{i=1}^{K} c_i x_i, \quad \text{subject to} \quad \sum_{i \in S_j} x_i = 1, \quad j = 1, \ldots, n, \tag{2}
\]

\[
\sum_{i \in Y_h} x_i \leq |Y_h| - 1, \quad h = 1, \ldots, a, \tag{3}
\]

\[
\sum_{i=1}^{K} x_i \leq m_0, \tag{4}
\]

\[
x_i \in \{0, 1\}, \quad i = 1, \ldots, K. \tag{5}
\]

“Set partitioning” constraints (2) ensure that each operation is executed on exactly one selected LF-station. “Cycle” constraints (3) ensure that the subgraph of graph \(G_w\), whose vertices are the selected LF-stations, is acyclic, which implies that all the selected stations can be ordered so that precedence relations for operations are fulfilled. Constraints (4) ensure that the number of selected LF-stations is at most \(m_0\).

Proposition 1. Vector \(x\) satisfies constraints (2)-(5) if and only if LF-stations of the corresponding unordered transfer line \(UTL(x)\) can be ordered to form a feasible transfer line.

Proof: Part “if” is trivial. Consider part “only if”, i.e., assume that \(x\) satisfies (2)-(5). Let the corresponding unordered transfer line be \(UTL(x) = \{W_{r_1}, \ldots, W_{r_m}\}\). Since the subgraph of graph \(G_w\), whose vertices are \(W_{r_1}, \ldots, W_{r_m}\), is acyclic due to (3), these vertices can be always topologically ordered. To simplify the notation, let \(L(x) = (W_{r_1}, \ldots, W_{r_m})\) be an arbitrary sequence of these vertices feasible with respect to \(G_w\). Sequence \(L(x)\) is a feasible transfer line. Indeed, the inclusion and exclusion constraints are satisfied for it because \(W_{r_1}, \ldots, W_{r_m}\) are LF-stations. Furthermore, each operation is executed exactly once due to the constraints (2). The precedence constraints are satisfied because for every pair of operations \(i\) and \(j\) such that \(i \rightarrow j\) and \(i \in \text{Oper}(W_{r_i}), j \in \text{Oper}(W_{r_j})\), we have \(h \neq g\) due to (2), and \(h < g\) due to the topological feasibility of the sequence \(L(x)\).

We deduce that the following two-stage procedure can be used to solve problem P. In the first stage, find an optimal solution, \(x^*\), of the problem MinSPP, and determine an optimal collection of LF-stations, \(\{W_{r_1}, \ldots, W_{r_m}\}\), such that \(\{r_1, \ldots, r_m\} = \{i \mid x_i^* = 1, i = 1, \ldots, K\}\). In the second stage, find an optimal sequence, \(L^*\), of these stations, which is an arbitrary sequence feasible with respect to the graph \(G_w\).

3. PREPROCESSING

The aims of our preprocessing procedure for problem P are the following: 1) reducing the set of blocks, 2) reducing the set of LF-stations, 3) determining LF-stations which are present in an optimal solution (definite LF-stations), and 4) tightening inequalities (3) and (4).

Extending precedence relations. The main instrument of the preprocessing procedure is an analysis of the precedence relations. If the set of arcs \(A_o\) of the precedence graph \(G_o = (N, A_o)\) is large, there is a good opportunity for reducing the set of blocks \(B\) and the set of LF-stations \(W\) without losing an optimal solution. Set \(A_o\) can be extended as follows. If operation \(i\) precedes operation \(j\) \(((i, j) \in A_o)\) and \(i\) belongs to an inclusion set \(I'\) \((i \in I' \in I\) and \(a\)[(i, j) such that \(k \in I'\) are added to \(A_o\). Similarly, if \((i, j) \in A_o\) and \(j \in I' \in I\), then arcs \((i, k)\) such that \(k \in I'\) are added to \(A_o\). After all such arcs have been added, opportunities for adding transitive arcs to the graph \(G_o\) may appear, which are realized.

Reducing the set \(B\) of blocks. We introduce an undirected graph \(G_b = (B, E_b)\) of block conflicts, in which the set of vertices coincides with the set of blocks \(B\) and there is a conflict edge \(\{b_1, b_2\} \in E_b\) if and only if blocks \(b_1 \in B\) and \(b_2 \in B\) have a common operation or there are operations \(i \in b_1, j \in b_1\) and \(k \in b_2, l \in b_2\) such that \(i\) precedes \(k\) and \(l\) precedes \(j\) \(((i, k) \in A_o, (l, j) \in A_o)\). Here vertices \(i\) and \(j\) may coincide or vertices \(l\) and \(k\) may coincide. Blocks of a conflict edge are called (mutually) conflicting. It is clear that conflicting blocks cannot appear in a feasible solution of MinSPP. The set of conflict edges is extended as follows. Consider blocks \(b_1\) and \(b_2\) such that \(\{b_1, b_2\} \not\in E_b\). Consider set \(B\) minus all blocks conflicting with \(b_1\) or \(b_2\). If this set does not cover the set \(N\) of all operations, then edge \(\{b_1, b_2\}\) is introduced. Again, in this case both \(b_1\) and \(b_2\) cannot appear in a feasible solution.

The set of blocks can be reduced as follows. If the set \(N\) of all operations is not covered by some block \(b\) and the blocks, which are not conflicting with it, then \(b\) cannot be present in any feasible solution, and it is removed from the set \(B\). An initial set \(W\) of LF-stations is constructed based on the reduced set \(B\).

Reducing the set \(W\) of LF-stations. If an LF-station contains at least two conflicting blocks, it is removed from the set \(W\). If there are two LF-stations \(W_i\) and \(W_j\) such that their sets of operations coincide and \(\sum_{b \in W_i} q(b) \leq \sum_{b \in W_j} q(b)\), then station \(W_j\) is removed from \(W\). For the remaining LF-stations, perform the following procedure. Graph \(G_w\) is constructed based on the reduced set \(W\). Two stations \(W_i\) and \(W_j\) are called (mutually) conflicting if they have conflicting blocks or there are arcs \((W_i, W_j)\) and \((W_j, W_i)\) in the graph \(G_w\). Conflicting stations cannot be present in a feasible solution of MinSPP. If the set \(N\) of all operations is not covered by some LF-station \(W\) and the LF-stations, which are not conflicting with it, then \(W\) cannot be present in any feasible solution, and it is removed from the set \(W\).

The sets \(S_j\) of LF-stations of operations \(j\), \(j \in N\), are constructed based on the reduced set \(W\). After this, if \(k \neq j\) and set \(S_k\) includes set \(S_j\) as a proper subset, then LF-stations of the set \(S_k\) are removed from \(W\).

Definite LF-stations. If some operation \(j\) belongs to a single LF-station \(W_j\), i.e., \(S_j = \{W_j\}\), then \(W_j\) is present in an optimal solution, and we set \(x_i = 1\).
Tightening inequalities (3) and (4). Let \( Q \) be a collection of non-intersecting sets \( S_j \) including the case of a single set \( S_j \). Inequality (4) can be tightened so that variables \( x_i \) corresponding to the LF-stations of \( Q \) are removed from it (but not from the other constraints and not from the objective function), and the right hand side bound is re-set as \( m_0 := m_0 - |Q| \).

A similar procedure can be executed for “cycle” inequalities (3): if LF-stations of a collection \( Q \) defined above all belong to a certain cycle \( Y_b \), then the corresponding inequality (3) can be tightened so that variables \( x_i \) corresponding to the LF-stations of \( Q \) are removed from it (but not from the other constraints and not from the objective function), and the right hand side bound is re-set as \( |Y_b| - 1 := |Y_b| - 1 - |Q| \).

All the preprocessing actions described above but the last are repetitive. For tightening one of the inequalities (3) or inequality (4), only one set \( Q \) (even if it is a single set \( S_j \)) can be used.

An application of the preprocessing actions “definite LF-stations” and “tightening inequalities (3) and (4)” can lead to a situation that in the MinSPP formulation an equality appears in which the right hand side contains zero. In this case, all variables in its left hand side are set to zero, which corresponds to removing associated LF-stations from the set \( \mathcal{W} \).

Preprocessing procedure is performed chronologically in the following way:

1. Extend precedence relations as it is described above.
2. Identify all possible pairs of conflicting blocks.
3. Apply the following algorithm to reduce the set blocks:
   a. Take a block \( b \).
   b. Remove all the blocks in conflict with \( b \) from the set of blocks \( B \).
   c. Verify if all operations can be performed by this block and the remaining ones. If not, then the block \( b \) cannot be present in any feasible solution and it therefore is removed.
4. Generate all possible combinations of blocks (LF-stations) such that an upper bound on the number of blocks per station is not exceeded, there are no conflicting blocks in one station, inclusion and exclusion contraints are not violated.
5. Apply the algorithm to reduce the set of LF-stations:
   a. Take a LF-station \( W \).
   b. Remove all LF-stations conflicting with \( W \) from the set of LF-stations \( \mathcal{W} \).
   c. Check if all operations can be covered by this LF-station and the remaining ones. If not, then the LF-station \( W \) cannot be present in any feasible solution and it is removed.
6. Search for definite stations LF-stations and apply tightening inequalities as it is described above.

4. COMPUTER EXPERIMENTS

Computer experiments were conducted on benchmark data sets for problem P in Belmokhtar [2006]. We used the commercial solver ILOG CPLEX 11.2 in this study. The experiments were run on PC with Intel Dual-Core 1.8 Mhz processor with 1 Gb of RAM under standard Linux environment.

When implementing the suggested solution approach for problem MinSPP, we encountered a memory problem, when the number of LF-stations, \( |\mathcal{W}| \), reaches tens of thousands and the number of cycles, \( a \), comes up to several millions. In order to handle instances of high dimension, the following iterative algorithm was introduced. Firstly, problem MinSPP is solved without the cycle constraints. There are five possible outputs of this solving:

1. no feasible solution is found due to the memory limit or computational time limit, which was set to 600 seconds;
2. a feasible solution with cycles is found;
3. an optimal solution with cycles is found;
4. a feasible solution without cycles is found;
5. an optimal solution without cycles is found.

Computations are terminated when the obtained solution has no cycles (cases (4) and (5)) or time or memory limit is exceeded (case (1)). In case (1), a lower bound for the optimal solution value of MinSPP is obtained. For cases (2) and (3), we add cycle constraints corresponding to cycles found, and run the algorithm again. In case (4), a feasible solution within a certain gap from the optimum is obtained. In case (5), an optimal solution is found.

The data sets in the computer experiments consist of 8 groups, where group \( i \) includes instances with \( 10(i + 1) \) operations, \( i = 1, \ldots, 8 \). Each group is subdivided into three series, for example, 20_1, 20_2 and 20_3, with low, medium, and high density of precedence constraints, respectively. Each series comprises 50 test instances.

Any instance with the high density of precedence constraints has at most three thousands LF-stations and it is easily solved to optimality by the described algorithm. No difficulty was also observed with the instances including up to 50 operations irrespectively of the density of precedence constraints.

Table 1 displays an evaluation of the number of LF-stations obtained after the preprocessing procedure for the most difficult series. Each entry of this table contains the number of instances of a series defining the row which have the number of LF-stations in the interval defining the column. The last row indicates the number of instances for which feasible solutions were found. In the overwhelming majority of the cases, the optimality was proved, except for 13 instances. For these instances the average relative deviation from the optimal solution values was equal to 1.4%, while the maximal deviation was 3.3%.

Instances with less than 20 thousands LF-stations can be considered as easy because all of them were solved to optimality. Most of the instances with up to 100 thousands LF-stations were solved to optimality as well. Some of them remained unsolved due to the computational time or memory limits. Instances with more than 100 thousands LF-stations are hard. Identification of conflicts and reducing the number of LF-stations was not performed for them due to the memory limit. About one third of these instances were solved, while for the major part only lower bounds were obtained.
Table 1. Number of LF-stations after preprocessing

<table>
<thead>
<tr>
<th>Series</th>
<th>&lt; 20000 LF-stations</th>
<th>[20000,100000] LF-stations</th>
<th>&gt; 100000 LF-stations</th>
</tr>
</thead>
<tbody>
<tr>
<td>60.1</td>
<td>10</td>
<td>36</td>
<td>4</td>
</tr>
<tr>
<td>60.2</td>
<td>50</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>70.1</td>
<td>6</td>
<td>34</td>
<td>10</td>
</tr>
<tr>
<td>70.2</td>
<td>49</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>80.1</td>
<td>2</td>
<td>30</td>
<td>18</td>
</tr>
<tr>
<td>80.2</td>
<td>48</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>90.1</td>
<td>0</td>
<td>19</td>
<td>31</td>
</tr>
<tr>
<td>90.2</td>
<td>39</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>204</td>
<td>135</td>
<td>63</td>
</tr>
<tr>
<td>Solved</td>
<td>204</td>
<td>109</td>
<td>18</td>
</tr>
</tbody>
</table>

It can be noticed that the series with low density of precedence constraints require more solving time. The reason is that the low density of precedence constraints induces the greater number of LF-stations. Therefore, the number of sequencing possibilities grows as well.

5. CONCLUSIONS

The transfer line design problem P has been reduced to the set partitioning problem, MinSPP. The reduction is based on the concept of a locally feasible station. Several ideas of preprocessing to decrease the number of variables and the size of the feasible domain of problem MinSPP have been presented. Computer experiments with MinSPP by means of a commercial solver have been performed. The experimental results show that MinSPP provides good solutions even for large size problems. Another advantage of the new model is its simplicity for understanding and computer coding.

REFERENCES

