International Journal of Computer Engineering and Information Technology

VOL. 8, NO. 7, July 2016, 115–124
Available online at: www.ijceit.org
E-ISSN 2412-8856 (Online)

Optimizing Multiset Metagrammars. Multigrammatical Representation of Classical Optimization Problems

Igor Sheremet[1] and Igor Zhukov[2]

[1] Financial University under the Government of Russian Federation, Moscow, Russian Federation

[2] National Research Nuclear University Moscow Engineering Physical Institute, Moscow, Russian Federation

[1]iasheremet@fa.ru, [2]i.zhukov@inbox.ru

ABSTRACT

Optimizing multiset metagrammars (OMMG) are specific knowledge representation model developed for problem solving in the areas of systems analysis and optimization. Formal definitions of OMMG syntax and semantics were considered in previous paper. Presented material is dedicated to OMMG application to the most well-known classical optimization problems. Techniques of multigrammatical representation of the following problems is considered: shortest path, travelling salesman, maximal flow, optimal assignments, optimal pair matching, transportation, integer linear programming.

Keywords: *Multisets, Optimizing multiset metagrammars, Optimization problems, Integer linear programming.*

1. INTRODUCTION

Syntax and semantics of optimizing multiset metagrammars (OMMG) where defined in the first part of the presented paper. OMMG may be used for various problems solving in such a way that their basic constructions (multiobjects, rules, metarules, variables-multiplicities and filters) and aggregates are interpreted in terms of the specific class of problems while algorithm generating set of terminal multisets (TMS) described by concrete OMMG provides creation of the solutions set, so every TMS corresponds to one solution. Techniques of OMMG toolkit application to different problems and their families form optimizing multiset metagrammars pragmatics.

Perhaps, the best beginning of OMMG pragmatics consideration are well-known classical optimization problems. Their multigrammatical representation (formulation) in the rest past of the paper aims two goals: first of all, to describe techniques of OMMG utilization; and secondly, to establish compliance between various problems and OMMG subfamilies, that is necessary for the estimation of their hardware implementations basic features.

We shall consider techniques mentioned from simplest to more sophisticated and general problems.

2. SHORTEST PATH AND TRAVELLING SALESMAN PROBLEMS

Shortest path problem [1,2] is usually formulated as follows.

Consider weighted oriented graph $G \subset A \times A \times \boldsymbol{N}$ such that $\langle a_i, a_j, l_{ij} \rangle \in G$ is edge (arc) of length l_{ij} connecting nodes a_i and a_j ($\boldsymbol{N}$ is set of all positive integer numbers). Path p is sequence of edges connected by common nodes

$$p = \left\langle \begin{matrix} (a_0, a_{i_1}, l_{0i_1}), \dots, (a_{i_j}, a_{i_{j+1}}, l_{i_j i_{j+1}}), \dots \\ \dots, (a_{i_{k-1}}, a_{i_k}, l_{i_{k-1} i_k}) \end{matrix} \right\rangle \quad (1)$$

where a_0 is initial node, a_{i_k} is final node, and

$$L = l_{0i_1} + \dots + l_{i_j i_{j+1}} + \dots + l_{i_{k-1} i_k} \quad (2)$$

is length of this path. Problem is to find all paths from initial to final node with minimal length.

This problem may be represented by optimizing multigrammar (i.e. OMMG without unitary metarules) $S_G = \langle a_0, R_G, F_G \rangle$, where every edge $\langle a_i, a_j, l_{ij} \rangle \in G$ corresponds to unitary rule

$$a_i \to 1 \cdot a_j, 1 \cdot e_{ij}, l_{ij} \cdot e, \quad (3)$$

where, in turn, a_i and a_j are objects corresponding to graph G nodes, e_{ij} is object corresponding to edge connecting a_i and a_j, while object e is, in essence, length measurement unit. Filter F_G contains the only optimizing condition

$$e = min. \quad (4)$$

As it is easy to see, set, generated by S_G, includes multisets of the form $\{1 \cdot a_{i_k}, 1 \cdot e_{0i_1}, \dots, 1 \cdot e_{i_{k-1}i_k}, 1 \cdot e_{i_k 0}, L \cdot e\}$, each corresponding to the path $\{(a_0, a_{i_1}, l_{0i_1}), \dots, (a_{i_{k-1}}, a_{i_k}, l_{i_{k-1}i_k})\}$ with length L, which is sum of all l_{ij} having place in unitary rules entering generation chain beginning from the multiset $\{1 \cdot a_0\}$. Path as a sequence of edges is presented by multiobjects $1 \cdot e_{0i_1}, \dots, 1 \cdot e_{i_{k-1}i_k}$. Condition (4) provides selection of all paths which have minimal length. Set $\bar{V}_{s_G}$ includes not one but all such paths.

Example 1. Let $G = \{\langle a_0, a_1, 3\rangle, \langle a_0, a_2, 1\rangle, \langle a_1, a_3, 5\rangle, \langle a_1, a_2, 4\rangle, \langle a_2, a_3, 2\rangle\}$ (Fig. 1). Here a_0 is initial node, a_3 – final node.

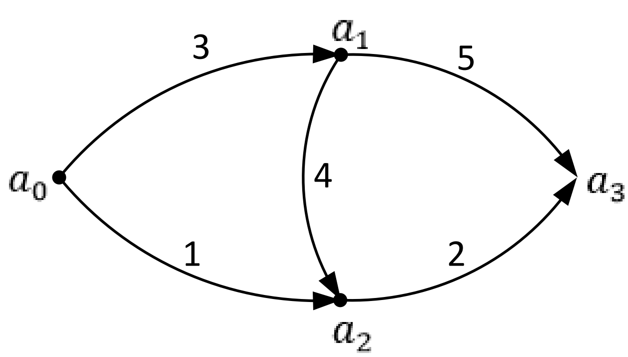

Fig. 1.

According to (3) scheme R_G contains following unitary rules:

$a_0 \to 1 \cdot a_1, 1 \cdot e_{01}, 3 \cdot e,$
$a_0 \to 1 \cdot a_2, 1 \cdot e_{02}, 1 \cdot e,$
$a_1 \to 1 \cdot a_3, 1 \cdot e_{13}, 5 \cdot e,$
$a_1 \to 1 \cdot a_2, 1 \cdot e_{12}, 4 \cdot e,$
$a_2 \to 1 \cdot a_3, 1 \cdot e_{23}, 2 \cdot e.$

As seen,

$$\bar{V}_{s_G} = \{\{1 \cdot a_3, 1 \cdot e_{02}, 1 \cdot e_{23}, 3 \cdot e\}\},$$

i.e. there is one shortest path of length 3 consisting of two edges ⟨0,2⟩ and ⟨2,3⟩. ■

Optimizing multigrammars (OMG) being subset of OMMG family as knowledge representation model provide easy and flexible formulation of various modifications of classical shortest path problem. For example, if every edge is marked not only by length which is directly associated with time for passing this edge, but also with necessary material resources spent while passing, then unitary rules entering R_G scheme would have form

$$a_i \to 1 \cdot a_j, 1 \cdot e_{ij}, l_{ij}^1 \cdot e_1, \dots, l_{ij}^m \cdot e_m, \quad (5)$$

where $e_1, \dots, e_m$ are mentioned resources measurement units while $l_{ij}^1, \dots, l_{ij}^m$ are their amounts necessary for corresponding edge passing. Filter may be

$$F_G = \{e = min, e_1 = min, \dots, e_m = min\}, \quad (6)$$

when one wants to minimize all resources amounts spent while path passing. If there is more soft formulation presuming resources limits available for traveler, then filter would be like

$$F_G = \{e = min, e_1 \le \bar{l_1}, \dots, e_m \le \bar{l}_m\}, \quad (7)$$

where $\bar{l_1}, \dots, \bar{l}_m$ are mentioned limits values. Of course, there may be threshold value for e unit while some of $e_1, \dots, e_m$ units values in TMS generated may be optimized. For example, e unit is "minute" while e_1 is "USD" or "EUR", then one may solve problem how to arrive to the final point not later than in $\bar{l}$ minutes spending for this minimal money. Then

$$F_G = \{e \le \bar{l}, e_1 = min\}.$$

As seen, there may be a lot of such combinations providing this kind of various multicriterial optimization problems formulation.

Travelling salesman problem (TSP) [2 – 4] is usually formulated as follows.

There is weighted oriented graph $G \subset A \times A \times \boldsymbol{N},$ where $\langle a_i, a_j, l_{ij}\rangle \in G$ has the same sense as higher. Difference is in consideration of only closed paths starting and ending at the same node a_0, containing all nodes $a \in A$, and in such a way that every node is visited once (such path is called Hamiltonian cycle). Problem is to find all Hamiltonian cycles having minimal length.

This problem may be represented by optimizing multigrammar $S_G = \langle a_0, R_G, F_G\rangle$, where every edge $\langle a_i, a_j, l_{ij}\rangle \in G$ corresponds to unitary rule

$$a_i \to 1 \cdot a_j, 1 \cdot e_j, 1 \cdot e_{ij}, l_{ij} \cdot e, \quad (8)$$

where a_j is non-terminal object serving for generation (i.e. node passing) continuation, e_j serves for j-th node number of visits accumulation, e_{ij} provides representation of edge connecting i-th and j-th nodes, while e is length measurement unit. All e_i, e_{ij} and e are terminal objects.

Filter F_G includes, as higher, one optimizing condition $e = min$ and, besides, $|A|$ boundary conditions

$$e_i = 1, \quad (9)$$

for all $i = 0, \dots, k,$ where $k = |A| - 1$. Condition $e = min$ provides selection of all paths with minimal length while conditions (9) – selection of those paths which contain all nodes, and every of them is visited only once. Due to (9) all paths which contain multiple node occurrences are eliminated.

As may be seen, every TMS generated by OMG S_G includes multisets of the form

$$\{1 \cdot e_0, 1 \cdot e_1, \ldots, 1 \cdot e_j, \ldots, 1 \cdot e_k, 1 \cdot e_{0i_1}, \ldots, 1 \cdot e_{i_{k-1}0}, L \cdot e\}, \quad (10)$$

each corresponding to its own closed path $a_0 \Rightarrow a_0$ with length L, and the last is sum of all multiplicities l_{ij} from URs applied while generation beginning from multiset $\{1 \cdot a_0\}$.

Example 2. Let $G = \{\langle a_0, a_1, 3\rangle, \langle a_1, a_3, 5\rangle, \langle a_1, a_2, 1\rangle, \langle a_2, a_0, 3\rangle, \langle a_3, a_2, 4\rangle\}$ (Fig. 2). Then according to (8) – (9) R_G

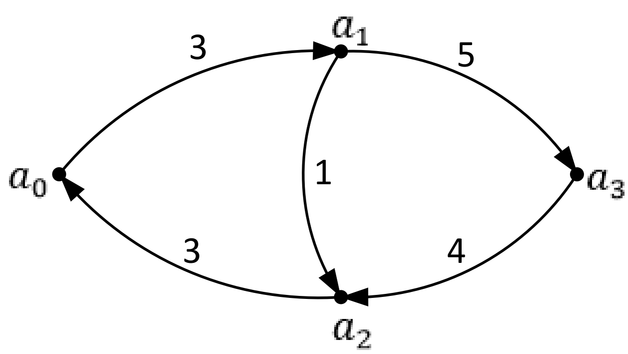

Fig. 2.

scheme will contain following unitary rules:

$$a_0 \to 1 \cdot a_1, 1 \cdot e_1, 1 \cdot e_{01}, 3 \cdot e,$$
$$a_1 \to 1 \cdot a_3, 1 \cdot e_3, 1 \cdot e_{13}, 5 \cdot e,$$
$$a_1 \to 1 \cdot a_2, 1 \cdot e_2, 1 \cdot e_{12}, 1 \cdot e,$$
$$a_2 \to 1 \cdot a_0, 1 \cdot e_0, 1 \cdot e_{20}, 3 \cdot e,$$
$$a_3 \to 1 \cdot a_2, 1 \cdot e_2, 1 \cdot e_{32}, 4 \cdot e.$$

Filter

$$F_G = \{e_0 = 1, e_1 = 1, e_2 = 1, e_3 = 1, e_4 = 1, e = min\}.$$

As seen,

$$\overline{V_{s_G}} = \{\{1 \cdot e_0, 1 \cdot e_1, 1 \cdot e_2, 1 \cdot e_3, 1 \cdot e_4, 1 \cdot e_{01}, 1 \cdot e_{13}, 1 \cdot e_{32}, 1 \cdot e_{24}, 15 \cdot e\}\}. \blacksquare$$

All the said higher while shortest path problem consideration is, of course, applicable to the travelling salesman problem. Let us, however, consider one more modification of the last called usually as “set TSP” or “generalized TSP” [4]. It is formulated as follows.

There are $m > 1$ travelling salesmen being located at nodes (points) $\overline{i_1}, \ldots, \overline{i_m}$. Starting from these points, they must visit all the rest nodes in such a way that every node must be visited once, all salesmen must return to their starting points, and total length of all salesmen paths must be minimal.

This problem may be represented by OMG $S_G = \langle a_0, R_G, F_G\rangle$, where, as in the classical TSP, every edge $\langle a_i, a_j, l_{ij}\rangle \in G$ corresponds to unitary rule

$$a_i \to 1 \cdot a_j, 1 \cdot e_j, 1 \cdot e_{ij}, l_{ij} \cdot e. \quad (11)$$

The only difference is that a_0 is not starting node, but title object being head of UR

$$a_0 \to 1 \cdot a_{\overline{i_1}}, \ldots, 1 \cdot a_{\overline{i_j}}, \ldots, 1 \cdot a_{\overline{i_m}}, \quad (12)$$

which provides generation start from objects corresponding to salesmen starting points.

All other components of S_G are just the same as in the classical TSP: optimizing condition is $e = min$, and boundary conditions are $e_{i_1} = 1$ for all $i = 0, 1, \ldots, |A| - 1$.

As may be seen, OMG in all various TSP modifications are cyclic, and if there would not be filters with conditions cutting off all redundant paths containing multiple visits of the same nodes, i.e. multisets with objects e_i, e_{ij} which multiplicities are greater then 1, these OMG would generate infinite sets of terminal multisets. Note also, that considered OMG, unlike classical grammars, may generate multisets containing title object a_0 (in classical grammars axioms occurrence in right parts of the generation rules is prohibited).

Evidently, it is sufficient to use optimizing multigrammars for shortest path and travelling salesman problems representation. Problems considered lower need application of more sophisticated toolkit, i.e. optimizing multiset metagrammars.

3. MAXIMAL FLOW PROBLEM

Maximal flow problem [5, 6] is usually formulated as follows.

There is transport network being weighted oriented graph $G \subset A \times A \times \boldsymbol{N}$ with the initial node $a_0 \in A$ (source) and final node $a_k \in A$ (sink). Here $\langle a_i, a_j, l_{ij}\rangle \in G$ means l_{ij} is flow capacity of the edge connecting a_i and a_j. Flow in network G is weighted oriented graph $\bar{G} \subset A \times A \times \boldsymbol{N}$ in which for every $\langle a_i, a_j, l_{ij}\rangle \in G$ there exists $\langle a_i, a_j, \overline{l_{ij}}\rangle \in \bar{G}$ such that $\overline{l_{ij}} \le l_{ij}$.

Backbone feature of transport network is so called flow conservation property written as follows:

$$\sum_{\langle a_i, a_j, \overline{l_{ij}}\rangle \in \bar{G}} \overline{l_{ij}} = \sum_{\langle a_j, a_r, \overline{l_{jr}}\rangle \in \bar{G}} \overline{l_{jr}}, \quad (13)$$

i.e. for every node a_j total input and total output flows are equal. As seen from the flow conservation property, total flow which runs from source a_0 is equal to total flow entering sink a_k:

$$\sum_{\langle a_0, a_i, \overline{l_{0j}}\rangle \in \bar{G}} \overline{l_{ij}} = \sum_{\langle a_j, a_k, \overline{l_{jk}}\rangle \in \bar{G}} \overline{l_{jk}} = L. \quad (14)$$

Maximal flow problem is to find all flows $\bar{G}$ with maximal $\bar{L}$ value from all possible flows.

Let us construct optimizing multimetagrammar $S_G = \langle a_0, R_G, F_G \rangle$ representing this problem.

We shall include to R_G scheme one unitary metarule

$$a_0 \to x_{0i_1} \cdot e_{0i_1}, \dots, x_{j_q k} \cdot e_{j_q k}, \tag{15}$$

which multiplicities-variables x_{ij} correspond to the unknown variables $\overline{l_{ij}}$ in classical formulation while e_{ij} are non-terminal objects corresponding to graph G edges. Also we shall include to R_G scheme $|G|$, by graph G edges quantity, unitary rules of the form

$$e_{ij} \to 1 \cdot e_i^-, 1 \cdot e_j^+, 1 \cdot \overline{e_{ij}}, \tag{16}$$

where e_i^- and e_j^+ are terminal objects, which multiplicities in the generated terminal multisets are total flows running from i-th node and entering j-th node correspondingly; $\overline{e_{ij}}$ is terminal object which multiplicity in generated TMS is flow passing from i -th to j-th node.

Filter F_G would contain $k-1$ boundary conditions of the form

$$e_i^+ = e_i^-, \tag{17}$$

where $i = 1, \dots, k-1$, which reflect equality of the input and output flows for every node excepting source and sink. The last are connected by one more boundary condition

$$e_0^- = e_k^+, \tag{18}$$

which defines equality of input and output flows of the whole network. And, at last, F_G would contain $|G|$ evident chain boundary conditions of the form

$$0 \le x_{ij} \le l_{ij}, \tag{19}$$

for all $\langle a_i, a_j, l_{ij} \rangle \in G$, as well as one optimizing condition

$$e_0^- = max. \tag{20}$$

As seen from the above description, all terminal multisets

$$v = \{n \cdot e_0^-, \dots, n_i^- \cdot e_i^-, n_{ij} \cdot e_{ij}, n_j^+ \cdot e_j^+, \dots, n \cdot e_k^+\} \in \overline{V_{S_G}}, \tag{21}$$

which satisfy (15) – (18), are solutions of the problem, and n is maximal flow.

Example 3. Let
$G = \{\langle a_0, a_1, 5\rangle, \langle a_1, a_3, 6\rangle, \langle a_1, a_4, 8\rangle, \langle a_3, a_4, 6\rangle, \langle a_0, a_2, 7\rangle, \langle a_2, a_4, 10\rangle\}$ (Fig. 3). Then according to (15) – (20),

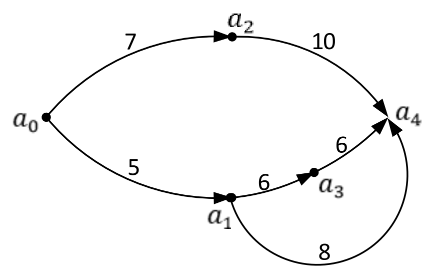

Fig. 3.

R_G scheme of the OMG $S_G = \langle a_0, R_G, F_G \rangle$ contains following unitary rules and metarules:

$$a_0 \to x_{01} \cdot e_{01}, x_{13} \cdot e_{13}, x_{14} \cdot e_{14}, x_{34} \cdot e_{34}, x_{02} \cdot e_{02}, x_{24} \cdot e_{04},$$
$$e_{01} \to 1 \cdot e_0^-, 1 \cdot e_0^+, 1 \cdot \overline{e_{01}},$$
$$e_{13} \to 1 \cdot e_1^-, 1 \cdot e_3^+, 1 \cdot \overline{e_{13}},$$
$$e_{14} \to 1 \cdot e_1^-, 1 \cdot e_4^+, 1 \cdot \overline{e_{14}},$$
$$e_{34} \to 1 \cdot e_3^-, 1 \cdot e_4^+, 1 \cdot \overline{e_{34}},$$
$$e_{02} \to 1 \cdot e_0^-, 1 \cdot e_2^+, 1 \cdot \overline{e_{02}},$$
$$e_{24} \to 1 \cdot e_2^-, 1 \cdot e_4^+, 1 \cdot \overline{e_{24}}.$$

F_G filter contains following boundary conditions:

$$e_1^+ = e_1^-, e_2^+ = e_2^-, e_3^+ = e_3^-, e_0^- = e_4^+,$$
$$0 \le x_{01} \le 5, 0 \le x_{02} \le 7, 0 \le x_{13} \le 6, 0 \le x_{14} \le 8,$$
$$0 \le x_{24} \le 10, 0 \le x_{34} \le 6.$$

Optimizing condition is $e_0^- = max$.

As may be seen,

$$\overline{V_{S_G}} = \{\{5 \cdot \overline{e_{01}}, 7 \cdot \overline{e_{02}}, 5 \cdot \overline{e_{13}}, 5 \cdot \overline{e_{34}}, 7 \cdot \overline{e_{24}}, 12 \cdot e_0^-, 5 \cdot e_1^+, 5 \cdot e_1^-, 7 \cdot e_2^+, 7 \cdot e_2^-, 5 \cdot e_3^+, 5 \cdot e_3^-, 12 \cdot e_4^+\},$$
$$\{5 \cdot \overline{e_{01}}, 5 \cdot \overline{e_{14}}, 7 \cdot \overline{e_{02}}, 7 \cdot \overline{e_{24}}, 12 \cdot e_0^-, 5 \cdot e_1^+, 5 \cdot e_1^-, 7 \cdot e_2^+, 7 \cdot e_2^-, 12 \cdot e_4^+\}\},$$

i.e. maximal flow in this network is 12, and there are two solutions of this problem.

By analogy with shortest path problem, maximal flow problem may be easily generalized if there is necessary to take into account costs of one flow unit transfer between every two nodes. If so, then every unitary rule of the form (16) may be transformed to

$$e_{ij} \to 1 \cdot e_i^-, 1 \cdot e_i^+, 1 \cdot \overline{e_{ij}}, c_{ij} \cdot e, \tag{22}$$

where c_{ij} is cost mentioned, while e is this cost measurement unit. After this it is sufficient to include additional boundary condition $e \le \bar{c}$, where $\bar{c}$ is available resource (for example, money), to filter F_G. As seen, this extension provides selection of those terminal multisets generated, which correspond to permissible cost of the flow transfer from source to sink. Selected TMS would have form

$$v = \{c \cdot e, n \cdot e_0^-, \dots, n_i^- \cdot e_i^-, n_{ij} \cdot e_{ij}, n_j^+ \cdot e_j^+, \dots, n \cdot e_k^+\} \in \overline{V_{S_G}}, \tag{23}$$

where c is cost of this variant of flow transfer.

If it is necessary to select variant with minimal cost, it is sufficient to include to F_G optimizing condition $e = min$ instead of $e \leq \bar{c}$. Of course, as in section 2, there may be more than one type of resources spent while passing network, so (22) body may contain $k > 1$ multiobjects $c_{ij}^1 \cdot e_1, \ldots, c_{ij}^k \cdot e_k$. Similary, there may be $l > 1$ optimizing conditions.

4. ASSIGNMENT, OPTIMAL PAIR MATCHING AND TRANSPORTATION PROBLEMS

Assignment problem [8, 9] is formulated usually as follows.

There is assignment matrix $X_{n\times n}$, where $x_{ij} = 1$, if i-th work is assigned to j-th person, $x_{ij} = 0$ otherwise. Cost matrix $C_{n\times n}$ contains information about costs of the assignments, so c_{ij} is cost of i-th work execution by j-th person. Problem is to assign works to persons in such a way that total cost defined as

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \quad (24)$$

would be minimal, while one person may execute only one work, and one work may be executed by one person only:

$$\sum_{i=1}^{n} x_{ij} = 1 \text{ for all } j = 1, \ldots, n, \quad (25)$$

$$\sum_{j=1}^{n} x_{ij} = 1 \text{ for all } i = 1, \ldots, n. \quad (26)$$

Optimizing multiset metagrammar $S =< problem, R, F >$ represents this problem in such a way that R scheme includes following unitary metarules:

$problem \rightarrow 1 \cdot opt, 1 \cdot strings, 1 \cdot columns, 1 \cdot solution,$ (27)

$opt \rightarrow c_{11} \cdot a_{11}, \ldots, c_{ij} \cdot a_{ij}, \ldots, c_{nn} \cdot a_{nn},$ (28)

$a_{11} \rightarrow x_{11} \cdot e,$

$\ldots$

$a_{ij} \rightarrow x_{ij} \cdot e,$ (29)

$\ldots$

$a_{nn} \rightarrow x_{nn} \cdot e,$

$strings \rightarrow x_{11} \cdot s_1, \ldots, x_{1n} \cdot s_1, \ldots, x_{ij} \cdot s_i, \ldots, x_{n1} \cdot s_n, \ldots, x_{nn} \cdot s_n,$ (30)

$columns \rightarrow x_{11} \cdot o_1, \ldots, x_{n1} \cdot o_1, \ldots, x_{ij} \cdot o_j, \ldots, x_{1n} \cdot o_n, \ldots, x_{nn} \cdot o_n,$ (31)

$solution \rightarrow x_{11} \cdot e_{11}, \ldots, x_{ij} \cdot e_{ij}, \ldots, x_{nn} \cdot e_{nn}.$ (32)

Here e is terminal object, which multiplicity in generated terminal multisets is value of the optimized function (24); $s_i (i = 1, \ldots, n)$ is terminal object corresponding to i-th string of X matrix, and this object multiplicity in the generated TMS is sum of x_{ij} values having place in this i-th string; o_j $(j = 1, \ldots n)$ is similar terminal object corresponding in the mentioned sense to j-th column of X. Multiplicities-variables x_{ij} correspond exactly to variables x_{ij} in the classical formulation (24)-(26). Terminal objects $e_{11}, \ldots, e_{nn}$ serve for solution direct representation.

It is quite evident, that F filter would contain following boundary conditions corresponding to classical formulation of the assignment problem:

$s_1 = 1, \ldots, s_n = 1, \; o_1 = 1, \ldots, o_n = 1,$ (33)

$0 \leq x_{11} \leq 1, \ldots, 0 \leq x_{ij} \leq 1, \ldots, 0 \leq x_{nn} \leq 1.$ (34)

Optimizing condition entering F is also obvious:

$e = min.$ (35)

Now one can see, that terminal multisets generated by OMG $S =< problem, R, F >$ have form

$$\{(\sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij}) \cdot e, 1 \cdot s_1, \ldots, 1 \cdot s_n, 1 \cdot o_1, \ldots, 1 \cdot o_n, x_{11} \cdot e_{11}, \ldots, x_{nn} \cdot e_{nn}\}, \quad (36)$$

where multiobjects $x_{ij} \cdot e_{ij}$ with zero multiplicities are eliminated, so number of multiobjects $1 \cdot e_{ij}$ in every such TMS is n.

Example 4. Consider assignment problem, which goal is

$$2x_{11} + 3x_{12} + 4x_{21} + x_{22} \rightarrow min.$$

This problem is represented by OMMG $S =< problem, R, F >$, where R includes following unitary metarules:

$opt \rightarrow 2 \cdot a_{11}, \; 3 \cdot a_{12}, \; 4 \cdot a_{21}, \; 1 \cdot a_{21},$

$a_{11} \rightarrow x_{11} \cdot e,$

$a_{12} \rightarrow x_{12} \cdot e,$

$a_{21} \rightarrow x_{21} \cdot e,$

$a_{22} \rightarrow x_{22} \cdot e,$

$strings \rightarrow x_{11} \cdot s_1, x_{12} \cdot s_1, x_{21} \cdot s_2, x_{21} \cdot s_2, x_{22} \cdot s_2,$

$columns \rightarrow x_{11} \cdot o_1, x_{21} \cdot o_1, x_{12} \cdot o_2, x_{22} \cdot o_2,$

$solution \rightarrow x_{11} \cdot e_{11}, x_{12} \cdot e_{12}, x_{21} \cdot e_{21}, x_{22} \cdot e_{22}$

as well as unitary rule (27). According to (33)-(34)

$F = \{s_1 = 1, s_2 = 1, \; o_1 = 1, \; o_2 = 1, 0 \leq x_{11} \leq 1, 0 \leq x_{12} \leq 1, 0 \leq x_{21} \leq 1, 0 \leq x_{22} \leq 1, e = min\}.$

From this

$\overline{V_S} = \{\{3 \cdot e, 1 \cdot e_{11}, 1 \cdot e_{22}, 1 \cdot s_1, 1 \cdot s_2, 1 \cdot o_1, 1 \cdot o_2\}\}.$

As higher, it is quite easy to generalize this problem by additional optimality criteria as well as by additional boundary conditions.

Optimal pair matching problem [8, 9] is usually formulated as follows.

Consider weighted non-oriented graph

$G = \{< i_1, j_1, c_{i_1 j_1} >, \ldots, < i_m, j_m, c_{i_m j_m} >\},$ (37)

where $< r, q, c_{rq} > \in G$ means that c_{rq} is weight of edge connecting nodes r and q. Pair matching is set

$M' \subseteq M = \{< i_1, j_1 >, \ldots, < i_m, j_m >\},$ (38)

such that no any two edges have common node:

$(\forall <i,j> \in M')(\forall <k,l> \in M')\{i,j\} \cap \{k,l\} = \{\emptyset\}$. (39)

Pair matching weight is sum of weights of all edges entering M', and optimal pair matching $\overline{M}$ is such one, which has minimal weight:

$$C_{M'} = \sum_{<i,j>\in M'} c_{ij}, \quad (40)$$

$$(\forall M' \neq \overline{M})C_{\overline{M}} \leq C_{M'}. \quad (41)$$

Optimizing multiset metagrammar $S_G =< a_0, R_G, F_G >$ representing this problem is defined as follows.

URs and UMRs entering R_G scheme contain non-terminal objects which are written in the similar form $[i,j]$, as $< i,j > \in M'$ in usual record. R_G includes one unitary rule

$$a_0 \to 1 \cdot [i_1, j_1], \dots, 1 \cdot [i_m, j_m], \quad (42)$$

where every edge $[i,j] \in M$ is represented by corresponding object $[i,j]$, and also R_G includes one UMR and one UR for each edge.

Unitary metarule corresponding to k-th edge is

$$[i_k, j_k] \to x_k \cdot cost_k, x_k \cdot [i_k * j_k], x_k \cdot [i_k], x_k \cdot [j_k], \quad (43)$$

where x_k is multiplicity-variable, which value defines if k-th edge belongs to pair matching (1 if yes, 0 otherwise), $[i_k]$ and $[j_k]$ are terminal objects corresponding to i_k and j_k nodes, which are incident to k-th edge, $[i_k * j_k]$ is also terminal object corresponding to edge $[i_k, j_k]$, and $cost_k$ is auxiliary non-terminal object serving for representation of k-th edge weight.

Unitary rule representing weight of the edge is

$$cost_k \to c_k \cdot e, \quad (44)$$

(c_k units e).

F_G filter includes boundary conditions

$$0 \leq x_k \leq 1, \quad (45)$$

$$[i_k] = 1, \quad (46)$$

for all $k = 1,\dots,m$, as well as one optimizing condition

$$e = max. \quad (47)$$

From the described it is clear, that terminal multisets generated by S_G are of the form

$$\{(\sum_{[i,j]\in M'} c_{ij}) \cdot e\} \cup (\bigcup_{[i,j]\in M'} \{1 \cdot [i*j], 1 \cdot [i], 1 \cdot [j]\}), \quad (48)$$

where terminal object e multiplicity is weight of the M' pair matching, while multiobjects of the form $1 \cdot [i * j]$ represent edges entering this pair matching. Note that boundary condition (46) excludes such TMS, which contain terminal multiobject like $l \cdot [i]$, where l]1, as well as TMS, where some of them are absent because of $l = 0$. So, (46) selects those only terminal multisets which correspond to (38)-(39) pair matching definition.

Example 5. Consider optimal pair matching problem with graph $G = \{[1,2,3], [1,3,4], [2,4,4], [3,4,2]\}$ (Fig. 4).

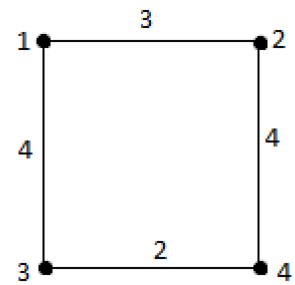

Fig. 4

According to (42) – (44), R_G scheme of the corresponding this problem optimizing multiset metagrammar $S_G = [a_0, R_G, F_G]$ contains following unitary rules and unitary metarules:

$a_0 \to 1 \cdot [1,2], 1 \cdot [1,3], 1 \cdot [2,4], 1 \cdot [3,4]$,
$[1,2] \to x_1 \cdot cost_1, x_1 \cdot [1], x_2 \cdot [3], x_2 \cdot [1*3]$,
$cost_1 \to 3 \cdot e$,
$[1,3] \to x_2 \cdot cost_2, x_2 \cdot [1], x_2 \cdot [3], x_2 \cdot [1*3]$,
$cost_2 \to 4 \cdot e$,
$[2,4] \to x_3 \cdot cost_3, x_3 \cdot [2], x_3 \cdot [4], x_3 \cdot [2*4]$,
$cost_3 \to 4 \cdot e$,
$[3,4] \to x_4 \cdot cost_4, x_4 \cdot [3], x_4 \cdot [4], x_4 \cdot [3*4]$,
$cost_4 \to 2 \cdot e$.

According to (45) – (47), F_G filter contains boundary conditions $0 \leq x_1 \leq 1$, $0 \leq x_2 \leq 1, 0 \leq x_3 \leq 1, 0 \leq x_4 \leq 1, [1] = 1, [2] = 1, [3] = 1, [4] = 1$ as well as optimizing condition e = max. From here

$$\overline{V_{S_G}} = \{\{8 \cdot e, 1 \cdot [1*3], 1 \cdot [2*4], 1 \cdot [1], 1 \cdot [2], 1 \cdot [3], 1 \cdot [4]\}\}$$

So optimal pair matching includes edges [1,3] and [2,4] and has weight 8.

Transportation problem [9, 10] representation techniques is very close to the described higher in this section. This problem is usually formulated as follows.

Find matrix $X_{m\times n}$ from the condition

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij} \to min, \quad (4$$

under restrictions

$$\sum_{j=1}^{n} x_{ij} = a_i, \quad i \in \{1,\dots,m\}, \quad (50)$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j \in \{1,\dots,n\}, \quad (51)$$

$$x_{ij} \geq 0, \quad i \in \{1,\dots,m\}, \ j \in \{1,\dots,n\}. \quad (52)$$

Here x_{ij} value is load amount transported from i-th manufacturer (source point) to j-th consumer (destination point) while c_{ij} is cost of transportation of mentioned load unit, so sum of products $c_{ij} \cdot x_{ij}$ by all possible values of i and j is cost of transportation of all x_{ij} amounts from manufactures to consumers. Condition (49) minimizes total expenses for this action, while a_i is total load amount at i-th source point and b_j is total load amount at j-th destination point.

Optimizing multiset metagrammar $S_G = < problem, R, F >$ representing (49)-(52) is as follows.

R scheme by analogy with (27) – (32) would contain following unitary rules and unitary metarules:

$$problem \to 1 \cdot opt, 1 \cdot sources, 1 \cdot destination \quad (53)$$

$$opt \to c_{11} \cdot e_{11}, \dots, c_{ij} \cdot e_{ij}, \dots, c_{mn} \cdot e_{mn}, \quad (54)$$

$$sources \to x_{11} \cdot \overline{a_1}, \dots, x_{1n} \cdot \overline{a_1}, \dots, x_{m1} \cdot \overline{a_m}, \dots, x_{mn} \cdot \overline{a_m}, \quad (55)$$

$$destination \to x_{11} \cdot \overline{b_1}, \dots, x_{m1} \cdot \overline{b_1}, \dots, x_{1n} \cdot \overline{b_n}, \dots, x_{mn} \cdot \overline{b_n}, \quad (56)$$

where $x_{ij} (i \in \{1, \dots, m\}, j \in \{1, \dots, n\})$ are multiplicities-variables corresponding to variables from (27) – (32) while $\overline{a_1}, \dots, \overline{a_m}, \overline{b_1}, \dots, \overline{b_n}$ are terminal objects corresponding to source ($\overline{a_i}$) and destination ($\overline{b_j}$) points. Also R would contain $m \times n$ unitary rules of the form

$$e_{ij} \to x_{ij} \cdot e, x_{ij} \cdot \overline{e_{ij}}, \quad (57)$$

where e is terminal object – load measurement unit, x_{ij} has the same sense as higher, and $\overline{e_{ij}}$ is terminal object denoting transportation from i-th source point to j-th destination point, so that presence of multiobject $l \cdot \overline{e_{ij}}$ in terminal multiset generated by S OMMG means l units of load must be transported from i-th source point to j-th destination point.

Filter F would contain $m + n$ boundary conditions

$$\overline{a_1} = a_1, \dots, \overline{a_m} = a_m, \quad (58)$$

$$\overline{b_1} = b_1, \dots, \overline{b_n} = b_n, \quad (59)$$

and $m + n$ boundary conditions

$$0 \le x_{11} \le \overline{N}_{11}, \dots, 0 \le x_{1n} \le \overline{N}_{1n}, (60)$$

$$0 \le x_{m1} \le \overline{N}_{m1}, \dots, 0 \le x_{mn} \le \overline{N}_{mn}, \quad (61)$$

where

$$\overline{N}_{ij} = \min\{a_i, b_j\}, \quad (62)$$

because x_{ij} domain is intersection of intervals $[0, a_i]$ and $[0, b_i]$, corresponding to maximal values of x_{ij} in (50) and (51), which, in turn, correspond to minimal (i.e. zero) values of all the rest variables in the restriction. Also F would contain one optimizing condition

$$e = min. \quad (63)$$

As it easy to see now, solution of the problem in the multiset formulation is set $\overline{V_G}$ containing terminal multisets of the form

$$\{[\sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij} \cdot \overline{x_{ij}})] \cdot e\} \cup (\bigcup_{i=1}^{m} \bigcup_{j=1}^{n} \{\overline{x_{ij}} \cdot \overline{e_{ij}}\}) \cup \quad (64)$$
$$\cup \{a_1 \cdot \overline{a_1}, \dots, a_m \cdot \overline{a_m}, b_1 \cdot \overline{b_1}, \dots, b_n \cdot \overline{b_n}\},$$

where the first one-element multiset of the join is $\{\bar{C} \cdot e\}$, $\bar{C}$ being minimal total cost of transportation; the second multiset is distribution of load amounts in such a way that $\overline{x_{ij}}$ units are transported from i-th source point to j-th destination point, while the third multiset contains terminal multiobjects which multiplicities are total load amounts taken from source (a_i) and received by destination (b_j) for all $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, n\}$. Multiplicities $\overline{x_{ij}}$ are values of corresponding variables x_{ij} from classical formulation of the problem, i.e. they reflect its solution.

Example 6. Consider matrix

$$C = \begin{Vmatrix} 2 & 4 & 1 \\ 3 & 7 & 5 \end{Vmatrix},$$

which defines transport expenses of two manufacturers on their production transportation to three consumers. Manufactures have 12 and 8 production units while consumers need 6, 9 and 5 units respectively. Classical formulation of this problem is as follows:

$$2 \cdot x_{11} + 4 \cdot x_{12} + x_{13} + 3 \cdot x_{21} + 7 \cdot x_{22} + 5 \cdot x_{23} \to min$$

under restrictions

$$x_{11} + x_{12} + x_{13} = 12,$$
$$x_{21} + x_{22} + x_{23} = 8,$$
$$x_{11} + x_{21} = 6,$$
$$x_{12} + x_{22} = 9,$$
$$x_{13} + x_{23} = 5,$$
$$x_{11} \ge 0, x_{12} \ge 0, x_{13} \ge 0, x_{21} \ge 0, x_{22} \ge 0, x_{23} \ge 0.$$

Corresponding optimizing multiset metagrammar $S = < a_0, R, F >$ has R scheme, which includes following unitary rules and metarules:

$$a_0 \to 1 \cdot opt, 1 \cdot sources, 1 \cdot destination,$$
$$opt \to 2 \cdot e_{11}, 4 \cdot e_{12}, 1 \cdot e_{13}, 3 \cdot e_{21}, 7 \cdot e_{22}, 5 \cdot e_{23},$$
$$sources \to x_{11} \cdot \overline{a_1}, x_{12} \cdot \overline{a_1}, x_{13} \cdot \overline{a_1}, x_{21} \cdot \overline{a_2}, x_{22} \cdot \overline{a_2}, x_{23} \cdot \overline{a_2},$$
$$destination \to x_{11} \cdot \overline{b_1}, x_{21} \cdot \overline{b_1}, x_{12} \cdot \overline{b_2}, x_{22} \cdot \overline{b_2}, x_{13} \cdot \overline{b_3}, x_{23} \cdot \overline{b_3},$$
$$e_{11} \to x_{11} \cdot e, x_{11} \cdot \overline{e_{11}},$$
$$e_{12} \to x_{12} \cdot e, x_{12} \cdot \overline{e_{12}},$$
$$e_{13} \to x_{13} \cdot e, x_{13} \cdot \overline{e_{13}},$$
$$e_{21} \to x_{21} \cdot e, x_{21} \cdot \overline{e_{21}},$$
$$e_{22} \to x_{22} \cdot e, x_{22} \cdot \overline{e_{22}},$$
$$e_{23} \to x_{23} \cdot e, x_{23} \cdot \overline{e_{23}}.$$

F filter includes boundary conditions

$$\overline{a_1} = 12,\ \overline{a_2} = 8,\ \overline{b_1} = 6,\ \overline{b_2} = 9,\ \overline{b_3} = 5,$$
$$0 \le x_{11} \le 6,\ 0 \le x_{12} \le 9,\ 0 \le x_{13} \le 5,$$
$$0 \le x_{21} \le 6,\ 0 \le x_{22} \le 8,\ 0 \le x_{23} \le 5,$$

as well as optimizing condition

$e = min.$

Solution of this problem is one-element set

$\overline{V_S} = \{\{79 \cdot e, 6 \cdot \overline{e_{11}}, 3 \cdot \overline{e_{12}}, 3 \cdot \overline{e_{13}}, 6 \cdot \overline{e_{22}}, 2 \cdot \overline{e_{23}}\}\}$,

where absence of terminal multiobject $\overline{e_{21}}$ in the TMS $v \in \overline{V_S}$ means $x_{21} = 0$.

All of the three problems considered in this section may be modified and generalized by correcting optimizing multiset metagrammars presenting them; some of techniques for such correction was described in previous sections.

Having some experience in OMMG representation of the simply formulated optimization problems, we may consider now one more general problem – integer linear programming (ILP).

5. INTEGER LINEAR PROGRAMMING

ILP problem is formulated as follows [9-11].

Find vector $\|x_1 \dots x_n\|$ from the condition

$$\sum_{i=1}^{n} c_i x_i \rightarrow max$$

under constraints

$$\sum_{j=1}^{n} a_{1j} \cdot x_j \leq b_1,$$

$$\sum_{j=1}^{n} a_{mj} \cdot x_j \leq b_m, \quad (66)$$

$x_i \geq 0, i \in \{1, \dots, m\}$. (67)

We shall consider only ILP with $a_{ij} > 0, e_i > 0, i \in \{1, \dots, m\}, j \in \{1, \dots, n\}$.

Corresponding optimizing multiset metagrammar $S = < problem, R, F >$ may be constructed very closely to what was described higher.

R scheme would contain following elements:

$problem \rightarrow 1 \cdot opt, 1 \cdot conditions, 1 \cdot solution$, (68)

$opt \rightarrow c_1 \cdot y_1, \dots, c_n \cdot y_n,$ (69)

$conditions \rightarrow a_{11} \cdot n_{11}, \dots, a_{1n} \cdot n_{1n}, \ , a_{m1} \cdot n_{m1}, a_{mn} \cdot n_{mn}$, (70)

$solution \rightarrow x_1 \cdot \overline{x_1}, \dots, x_n \cdot \overline{x_n},$ (71)

as well as n unitary metarules

$y_j \rightarrow x_j \cdot e,$ (72)

where $j \in \{1, \dots, n\}$, and $m \times n$ unitary metarules

$u_{ij} \rightarrow x_j \cdot r_i,$ (73)

where $i \in \{1, \dots, m\}, j \in \{1, \dots, n\}$.

Here in (69) – (73) $c_1, \dots, c_n, a_{11}, \dots, a_{1n}, \dots, a_{m1}, \dots, a_{mn}$ are multiplicities – constants identical to constants having place in the classical formulation (65) – (67); $x_1, \dots, x_n$ are multiplicities-variables identical to variables having place in (69) – (73); terminal object e is goal function measurement unit; terminal objects $r_1, \dots, r_n$ are linear functions from (66) restrictions measurement units; $\overline{x_1}, \dots \overline{x_n}$ are terminal objects, which multiplicities in terminal multisets generated are $x_1, \dots, x_n$ variables values, so $\overline{x_1}, \dots \overline{x_n}$ serve for solutions representation. All the rest objects having place in (68) – (73) are intermediate non-terminal objects serving for TMS generation purposes. As may be seen from (68) – (73), TMS generated have the following form:

$\{c \cdot e, k_1 \cdot \overline{x_1}, \dots, k_n \cdot \overline{x_n}, c_1 \cdot r_1, \dots, c_m \cdot r_m\},$ (74)

where

$$c = \sum_{i=1}^{n} c_i \cdot k_i, \quad (75)$$

$$c_1 = \sum_{j=1}^{n} c_{1i} \cdot k_i,$$

$$\dots \quad (76)$$

$$c_m = \sum_{j=1}^{n} c_{mi} \cdot k_i,$$

and $k_1, \dots, k_n$ are mentioned values of variables $x_1, \dots, x_n$.

From (74) – (76) it is obvious that F filter of the constructed OMMG would contain $m + n$ boundary conditions

$$r_1 \leq b_1, \quad \dots \quad r_m \leq b_m, \quad (77)$$

$$0 \leq x_1 \leq l_1, \quad \dots \quad 0 \leq x_n \leq l_n, \quad (78)$$

as well as one optimizing condition

$e = min,$ (79)

where

$l_j = min \{E\left(\frac{b_1}{a_{1j}}\right), \dots, E\left(\frac{b_m}{a_{mj}}\right)\}$ (80)

is maximal value of x_j variable; this value corresponds to the case, where all other variables values are zero; because x_j has place in all inequalities (66), its domain would be intersection of domains each corresponding to its own inequality, i.e.

$$[0, l_j] = \bigcap_{i=1}^{m} \left[0, E\left(\frac{b_{ij}}{a_{ij}}\right)\right] = \left[0, \min\left\{E\left(\frac{b_1}{a_{1j}}\right), \dots, E\left(\frac{b_m}{a_{mj}}\right)\right\}\right], \quad (81)$$

where $E(\beta)$ is maximal integer number from all such numbers, which are not greater than rational number β.

Of course, there is more simple way for l_i definition – for example, $l_i = MAX$, where MAX is sufficiently large maximal integer number implemented. However, (79) reduces the search space essentially (like (62) in transportation problem), that's why it is most comprehensive from the computational complexity point of view.

Let us illustrate the said by the example.

Example 7. Consider following ILP:

$$2 \cdot x_1 + 3 \cdot x_2 + 6 \cdot x_3 \to max$$

under constraints

$$3 \cdot x_1 + 5 \cdot x_2 + 7 \cdot x_3 \leq 15,$$
$$12 \cdot x_1 + 2 \cdot x_2 + x_3 \leq 20,$$
$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0.$$

According to (68) – (72), optimizing multiset metagrammar $S =< problem, R, F >$ corresponding to this ILP contains following URs and UMRs as well as conditions:

$problem \to 1 \cdot opt, 1 \cdot conditions, 1 \cdot solution,$
$opt \to 2 \cdot y_1, 3 \cdot y_2, 6 \cdot y_3, x_1 \cdot \overline{x_1}, x_2 \cdot \overline{x_2}, x_3 \cdot \overline{x_3},$
$conditions \to 3 \cdot u_{11}, 5 \cdot u_{12}, 7 \cdot u_{13}, 12 \cdot u_{21}, 2 \cdot u_{22}, 1 \cdot u_{23},$
$y_1 \to x_1 \cdot e,$
$y_2 \to x_2 \cdot e,$
$y_3 \to x_3 \cdot e,$
$u_{11} \to x_1 \cdot r_1,$
$u_{12} \to x_2 \cdot r_1,$
$u_{13} \to x_3 \cdot r_1,$
$u_{21} \to x_1 \cdot r_2,$
$u_{22} \to x_2 \cdot r_2,$
$u_{23} \to x_3 \cdot r_2,$
$r_1 \leq 15,$
$r_2 \leq 20,$
$0 \leq x_1 \leq 1,$
$0 \leq x_2 \leq 3,$
$0 \leq x_3 \leq 2,$
$e = min.$

Solution is one-element set

$$\overline{V_s} = \{\{11 \cdot e, 1 \cdot \overline{x_1}, 1 \cdot \overline{x_2}, 1 \cdot \overline{x_3}, 15 \cdot r_1, 15 \cdot r_2\}\},$$

that corresponds to $x_1 = 1, x_2 = 1, x_3 = 1$ in the classical formulation.

Evidently, it is quite simple to represent multicriterial ILP replacing (68) by

$$problem \to 1 \cdot opt_1, \ldots, 1 \cdot opt_k, 1 \cdot conditions, 1 \cdot solution, \quad (82)$$

and replacing (69) by $l > 1$ unitary metarules

$$opt_i \to c_1^i \cdot y_1^i, \ldots, c_{n_i}^i \cdot y_{n_i}^i, \quad (83)$$

where $i \in \{1, \ldots, k\}$.

6. CONCLUSION

Compliance between optimization problems considered and classes of optimizing multimetagrammars providing their correct multigrammatical representation ("emulation") is accumulated in Table 2.

Table 2:

№	Optimization problem	Class of multiset metagrammars
1	Shortest path	OMG
2	Travelling salesman	OMG
3	Maximal flow	OMMG
4	Optimal pair matching	OMMG
5	Optimal assignments	OMMG
6	Transport	OMMG
7	Integer linear programming	OMMG

As shown in [12, 13], problem of generation of set of terminal multisets defined by optimizing multiset metagrammar is equivalent to the multicriterial problem of discrete polynomial programming. OMMG algoritmics (i.e. TMS generation procedures) is described in detail in [12, 13].

As one can see, optimizing multimetagrammars provide relatively simple and understandable representation of classical optimization problems, although we have demonstrated OMMG capabilities without any basic syntax and semantics refinements making classical problems representation more compact and natural. These refinements (first of all, so called composite multiobjects and composite multiplicities) will be considered in the following papers.

Along with such simplifications of OMMG, usual (vector-matrix etc.) representation of the problems considered higher may be used, as well as preprocessors translating these representations to the unified OMMG form. Note, however, that the last provides a lot of problems formulation not expressed in usual notation, and one of the main purposes of OMMG development was namely extension of expressional power of the optimization problems description by converging mathematical and logical programming.

Concerning directions of OMMG approach further development, we may point three most important of them:

1) OMMG pragmatics expansion to various practical problems from different areas;
2) OMMG basic "branches and bounds" generation algorithmics further improvement in order to minimize redundant steps number while TMS generation;
3) hardware/software implementation of TMS generation procedures including various unconventional computation paradigms application as well as specialized "multiset/multigrammatic computers" design.

There are many interest directions of OMMG pragmatics further development:

1) sociotechnical systems cooperation and confrontation/conflict modeling;
2) various products and services costs estimation;
3) business planning;
4) production supply chains and logistics modeling;
5) economical/industrial systems online optimal scheduling in volatile environment.

Up-to-date OMMG algorithmics software implementations [12-14] provide TMS generation branches cut-offs based on information having place in boundary and optimizing conditions. This algorithmics combines mixed computation and interval analysis techniques and may be improved by more deep and precise estimation of variables-multiplicities domains before generation steps execution.

“Brutal force” OMMG algorithmics implementation is based on parallel computers application. There may be three various approaches to TMS generation parallelization:

1) high-parallel environments created from general-purpose processors (this approach is purely software-intensive because of utilization of already existing hardware);
2) high-parallel environments created from special-purpose processors with reducted instructions set oriented to TMS generation;
3) implementation of constantly used OMMG (or its scheme) as a chip which may be digital or even analog device as was proposed long ago in [15] for polynomial mathematical programming problems solving.

Hardware implementation of OMMG approach further development is, perhaps, most attractive. Its background was mentioned in the introduction to the first part of the paper: various unconventional computational and programming paradigms including bio-inspired and quantum computing, chemical programming etc.
Author will be glad to discuss these directions with scholars working in close areas.

7. ACKNOWLEDGMENTS

Author is grateful to Prof. Noam Chomsky for words of support to author`s activity in chosen research direction. Special thanks to Prof. Jeffrey Ullman whose criticisms contributed to this paper content radical improvement. Author expands his appreciation to Prof. Igor Zhukov for help in the promotion of the presented paper.

REFERENCES

[1] Ahuja R.K., Magnati T.L., Orlin J.B. Network Flows: Theory, Algorithms and Applications. – Prentice Hall, 1993.
[2] Cherkassky B.V., Golberg A.V., Radzik T. Shortest Paths Algoritms: Theory and Experimental Evaluation. – Mathematical Programming. Ser. A, Vol. 73 (1996), №2.
[3] Papadimitriou C.H., Steiglitz K. Combinatorial Optimization: Algoritms and Complexity. – NY: Dover, 1998.
[4] Gutin G., Punnen A.P. The Travelling Salesman Problem and Its Variations. – Springer, 2006.
[5] Rego C., Gamboa D., Glover F., Osterman C. Travelling Salesman Problem Heuristics: Leading Methods, Implementations and Latest Advances. – European Journal of Operations Research, Vol. 211 (2011), No. 3.
[6] Lawler E. Combinatorial Optimization: Networks and Matroids. – Dover, 2001.
[7] Schrijver A. On the History of the Transportation and Maximum Flow Problems. – Mathematical Programming, Vol. 91 (2002), No. 3.
[8] Burkard R., Dell`Amico M., Martello S. Assignment Problems (Revised Report). – SIAM, 2012.
[9] Sigal I.H., Ivanova A.P. Introduction to Applied Discrete Programming: Models and Computational Algorithms. – Moscow: Physmathlit, 2002 (In Russian).
[10] Schrijver A. Combinatorial Optimization. – Berlin, New York: Springer-Verlag, 2003.
[11] Rao S.S. Engineering Optimization: Theory and Practice (4th ed.) – John Wiley & Sons, 2009.
[12] Sheremet I.A. Recursive Multisets and Their Applications – Moscow: Nauka, 2010 (in Russian).
[13] Sheremet I.A. Recursive Multisets and Their Applications – Berlin: NG Verlag, 2011.
[14] Karasev R.S., Sheremet I.A., An Approach to the Optimization of Algorithmical Apparatus of Recursive Multisets. – Information Technologies in Design and Manufacturing, 2013, №3.
[15] Dennis J.B. Mathematical Programming and Electrical Circuits. – Massachusetts Institute of Technology, 1959.