FRAMEWORK FOR AN INTELLIGENT FIELD OPERATION SYSTEM (IFOS) : PART 2 – TASK IDENTIFICATION AND PLANNING METHODOLOGY

Sung-Keun Kim[1] and Jeffrey S. Russell[2]

[1]*Ph.D. Candidate, University of Wisconsin, Madison, WI, U.S.A.* sung-keu@cae.wisc.edu
[2]*Professor, University of Wisconsin, Madison, WI, U.S.A.* russell@engr.wisc.edu

Abstract: This paper presents an automated task identification and planning methodology for IFOS described in Part 1. The task identification and planning in IFOS involve the translation of input data to the first-level tasks that may consist of more than one subtask, and the explanation of what will happen with the scheduled time zones, where it will happen, and how it will happen. The first-level tasks to be planned are highly dependent on the size of the construction site, the volume of field works, and the amount of construction equipment that is available.
Keywords: Intelligent system, Automated planning, Agent-based system, Field operation

1. INTRODUCTION

The authors have developed an intelligent system, called IFOS (Intelligent Field Operation System), for performing field operation tasks [1]. The main goals of IFOS are (1) to generate a plan automatically for construction equipment which performs field operations such as stripping, pushing, hauling, spreading, and compacting of soil or solid waste materials in a continuously changing environment, (2) to rationalize quality control corresponding to the execution by construction equipment, (3) to provide a seamless means of cooperation between construction equipment, and (4) to improve worker safety. This paper focuses on automated task identification and planning.

The task identification and planning in IFOS involve the translation of input data to the first-level tasks that may consist of more than one subtask, and the explanation of what will happen with the scheduled time zones, where it will happen, and how it will happen. The first-level tasks to be planned are highly dependent on the size of the construction site, the volume of field works, and the amount of construction equipment that is available.

The task planning processes are depicted in Figure 1. To achieve efficient task execution of equipment agents, a construction site is partitioned into several small work areas using the Quadtree algorithm. After partitioning, cut and fill volumes for each work area are calculated by comparing the existing topology with the proposed topology. Task objects are constructed to describe the characteristics of the given task to be performed by equipment agents in a certain work cell. A set of correlated task objects comprises a task package. It may be considered as an individual small project that is accomplished by stripping, hauling, spreading, and compacting materials to be handled. The process of determining which task package to assign first on an equipment mediator (EM) is called task priority sequencing. By applying heuristic rules, the priority of the task package can be determined effectively.

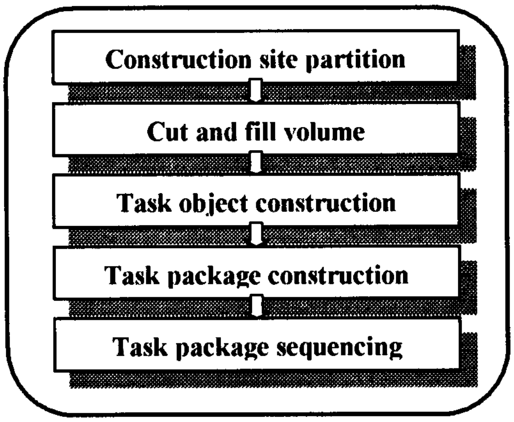

Figure 1. Process of Task Planning

2. TASK IDENTIFICATION

This section focuses on aspects of task identification, which is an approach to identify the tasks for field operation. To identify the task, the quantities of materials to be cut and filled should be determined from several input data (e.g., topological data, site condition data, quality requirement data, and weather data). Previous researches show that the earthwork volume can be evaluated by using a mass haul diagram that has long been used for the movement of materials or by applying heuristic and mathematical techniques. Previous methods, however, cannot be directly used for IFOS because they do not provide detailed information for field operation performed by IFOS' agents. In this section, four important issues are considered: (1) construction site partition, (2) task object, (3) task package, and (4) task package sequencing.

2.1 Construction Site Partition

Using the quadtree algorithm, which is a data structure based on recursive decomposition, the construction site can be partitioned into several work areas, called *work cells*, as shown in Figure 2. The size of a cell can be changed by the work volume that can be performed by one or a group of equipment within a certain period of time. In order to achieve more efficient task execution, the cell can be divided further into several small areas whose size is determined by the dimension of construction equipment. To represent work cell data, hierarchical data structure is used.

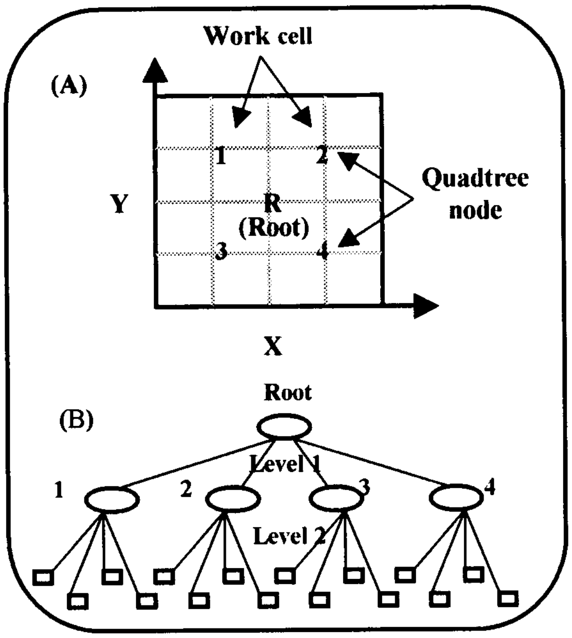

Figure 2. Example of Construction Site Partition

After partitioning the construction site, the cut and fill volumes for each cell are calculated from input data. The volume calculation is easily achieved by surface-to-surface comparison with cut and fill contours or surface to a planed plane. Every work cell is an object and has information on node number, node's color which represents the type of node, its coordination according to the entire construction site configuration, type of soil, cut and fill volumes, swell and shrinkage factors, and its parent, siblings, and children nodes (see Figure 3). The name of a data field is represented in "< >" and the types of data are shown in "| |".

The color of a cell shows the type of the cell. That means this coloring method is used for identifying the space characteristics efficiently. There are three available types of colors: *black*, *white*, and *gray*. A cell is said to be type *black* if its cut volume is larger than its fill volume, so it can provide other cells with soil. This cell can be also called a *positive cell*. A *white cell* means a *negative cell*, which needs soil to be filled, spread and compacted. The difference between its cut and fill volume should be transported into this cell. When the amount of cut volume is equal to that of fill volume, cut and fill volumes are zero, or earthwork has been done in a cell, this cell is a *gray cell*, called an *even cell*.

Work Cell

{
<Cell number> | Integer number |
<Color> | Black, White, Gray |
<Position> | Center (X_0,Y_0,Z_0), Upper right (X_1,Y_1,Z_1), Upper left (X_2,Y_2,Z_2), Lower right (X_3,Y_3,Z_3), Lower left (X_4,Y_4,Z_4) |
<Parent node> | Node number | // Pointer
<Sibling nodes> | Sibling work cell numbers | // Pointers
<Children node(*)> | Children node numbers | // Pointers
<Type of soil> | Sand, Rock, Clay, Common soil, Clay-sand and rock mixture |
<Soil consolidation> | Low, Medium, High |
<Swell factor> | Number |
<Shrinkage factor> | Number |
<Work volume> | Planed cut (P_Cut), Planed fill (P_Fill), Current cut (C_Cut), Current fill (C_Fill) |
}

Figure 3. Data Field of a Work Cell (Leaf Node)

The <Position> field has five coordinates: the center and four vertices of a work area. In the case of nodes that are not one of the leaf nodes, they have only center position information. The fields of <Parent>, <Siblings>, and <Children nodes> have pointers, which are used to access the location of indicated nodes. Information on children nodes is an option. When further decomposition is needed for effective and efficient task execution (e.g., equipment motion planning), the information on children nodes of a work cell is stored in the field of <Children node>. There are several data types of soil for the <Type of soil> field such as sand, rock clay, common soil, and so forth. Based on the soil type, the swell and shrinkage factors are changed. Planned and current work volumes are recorded in the slot of <Work volume>. As a default, planned cut and fill volumes are equal to current cut and fill volumes, respectively.

2.2 Task Object

The task object is an object that includes a specific task to be completed in a certain work cell, quality requirements, and the field operation level of detail (the second-level task) within a given task category. It has several data fields as shown in Figure 4, which describe the characteristics of a task to be performed by individual or a group of equipment. It is created when one of the work cells for the field operation is selected, and the type of the field operation is identified. The object is temporarily stored in an initial task object list. Each data field of a task object is summarized as follows:

Task Object List (First-level task)

········

Task Object (TO)

{
<TO number> | Integer number |
<What> | Earthwork or Landfill operation: Stripping, Pushing, Hauling, Spreading, and Compacting |
<Where> | Cell number | // Pointer
<Quality> | Density, Slope |
<Status> | Enumeration type: Initialized, Activated, Processing, and Done |
}

Figure 4. Data Field of a Task Object

The name of a data field is represented in "< >" and the types of data are shown in "| |". The <TO number> field has a unique integer number, which is assigned to a first-level task. The task type with related field operations is specified in the field of <What> which can have two kinds of possible work types for IFOS: (1) earthwork and (2) landfill operation, and five sorts of second-level tasks: (1) stripping, (2) pushing, (3) hauling, (4) spreading, and (5) compacting. The data field of <Where> represents a pointer that indicates a specific work cell. Using this pointer, cell information about node color, soil type, cut and fill volumes, etc. can be obtained. The quality requirements of task execution are shown in the field of <Quality>, and the status of task execution is recorded in the slot of <Status> as one of the enumeration types.

2.3 Task Package

A task package represents a set of correlated task objects. As shown in Figure 5, it consists of a source, at which soil to be transported is stripped, and one or more targets, at which the stripped soil is transported. So, it will require one or more task objects to be completed and these task objects are accomplished by stripping, pushing, hauling, spreading, and compacting. The task package keeps information on how much soil volume is transported where.

Task Package (TP)

{
<TP number> | Integer number |
<Source> | Task object number: Volume | // pointer
<Targets> | Task object number(s): Volume(s) | // pointer(s)
<Status> | Enumeration type: Initialized, Activated, Processing, and Done |
}

Figure 5. Data Field of a Task Package

Task packages are defended so as to expedite and enhance the information management for task planning and execution throughout the field operation process. It may be used as a control account for the purpose of collecting performance data for the master database. If needed, every task package may keep four kinds of information: (1) the early start time that is defined as the earliest time a task execution may begin, (2) the latest start time that is defined as the latest time a task execution could be started to finish it on time, (3) the time necessary to complete the first-level task, and (4) the time representing a deadline for the achievement of the given task.

There are five types of task package as shown in Figure 6. Basically, a task package consists of one or more task objects, a borrow pit, and/or a stockpile. The stripped soil is moved from a black cell to one or more white cell (Type 2). When there is no available white cell receiving soil any more, the soil is transported to the stockpile (Type 3 and Type 4). If there is no task object with a black cell in the activated list and there are one or more remaining task objects with a white cell, the soil is transported from a borrow pit to the white cell(s) (Type 5). Type 1 is a special case which means only one work cell is included in the task package. The stripped soil is moved from one portion of a cell to another portion of it. A series of operations such as stripping, transporting (pushing or hauling), spreading, and compacting can be modeled by a task package.

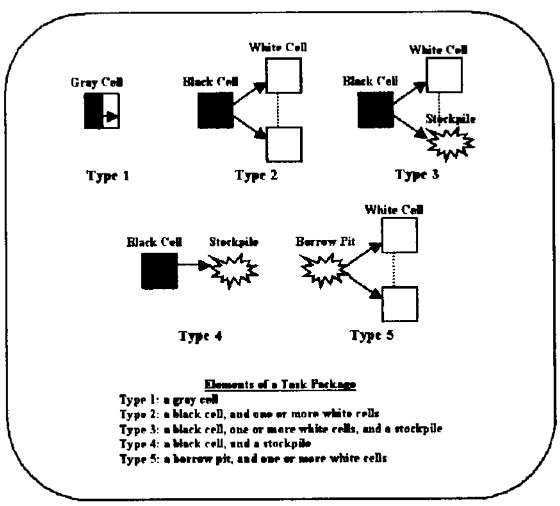

Figure 6. Five Types of Task Packages

2.4 Pre and Post-Task Object Selection

To construct a task package, a source and its target objects should be identified. After all task objects for the first-level task list are constructed, source and target relationship for every task object is established. The *source and target object selection algorithm* developed by the authors determines the sequence of steps for selecting source and target task objects for each task package. Basically, the proposed algorithm considers cut and fill volume balancing, and minimum hauling distance.

In the case of a task object with a black cell (a source object), it should search proper task object(s) with a negative cell to transport stripped soil based on the cut and fill volume distribution and haul distance. When one or more task objects are searched, they are considered as target objects, and are added into the target field of a task package. If all task objects with a black cell finish searching proper target objects and there are still remaining task objects that have a white cell, then task objects with a borrow pit are considered as their source task object. They are matched with the remaining task objects and are added into the source field of related task packages.

2.5 Pre and Post Task Object Search Rules

In order to minimize the distance of hauling or transporting soil, and the operation time, when a source object searches its target objects, it should find those which have a work cell located as near as possible, and this searching process should be done as soon as possible. To do so, a searching rule that is used in obtaining a searching sequence should be set.

There are two possible ways to determine target task objects which have a nearby work cell: (1) distance calculation between the centers of two cells, and (2) Quadtree searching. The first way is to calculate the distance between the work cell of searching task objects and that of searched task objects. When considering volume distribution requirements, choose appropriate task objects that are located in proximity to the source task object.

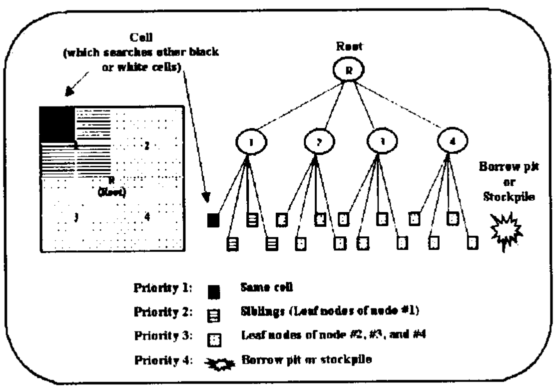

Figure 7. Example of Cell Searching Priority

Another way is to use the hierarchical information of the quadtree as shown in Figure 7. For better performance, only hierarchical information of the tree is used in searching target work cells. Basically, the nearer cells of a work cell are its siblings. In the quadtree data structure, siblings are on the same level in the hierarchy with the same parent node. The next nearest cells are the children nodes of the nodes that are on the same hierarchical level as its parent node, excluding its parent. The borrow pit or stockpile has the lowest priority for searching.

3. TASK PLANNING AND SCHEDULING

The main goal of task planning/scheduling is to identify the optimal sequence of task packages in real-time to achieve the global goals of IFOS, such as the fastest field operation completion and the maximization of equipment utilization. Since the amount of equipment is limited, the amount of equipment's idle time should be minimized, and the productivity of equipment should be maximized to achieve the maximum system efficiency and expedite project completion.

When task packages are scheduled with multiple equipment agent fleets to achieve the goals of IFOS, space constraints should be reflected when deciding the priority of task packages. If several equipment agents share the same physical space at the same time, this situation will result in the reduction of work productivity. When considering space interference between two task packages to be scheduled or between task package-in-progress and a task package to be scheduled, it is possible to achieve effective task scheduling that can expedite field operation completion.

Currently, four kinds of job scheduling approaches are used for job scheduling: (1) heuristic approaches or dispatching rules, (2) schedule permutation, (3) AI approaches, and (4) analytical methods [2]. For this research, the heuristic approach is used for the intelligent task planning and scheduling that are conducted in the multi-agent environment to identify the sequence of task packages for executing the required earthwork tasks and to allocate the available resources (e.g., equipment agents) to them. The best sequence of the task packages can be achieved using a series of heuristic rules.

3.1 Scheduling Heuristic Rules for Sequencing Task Package

Since the performance of a scheduling policy mainly depends on characteristics of a problem domain, there is a need to develop a scheduling heuristic procedure that reflects the special context of IFOS. This section describes TPSR (Task Package Sequencing Rule) that is a more problem-oriented procedure to resolve space interference between two or more groups of equipment to perform field operation tasks effectively.

Rule 1: No-Space Interference First (NSIF)

Every task package has a source and one or more target objects. If a target object is correlated with several sources, which means that stripped soil is transported from several white cells to fill the cell of the target object, and two or more equipment groups are involved at the same time to transport the stripped soil to the target cell, there will be space interference that will result in the decrease of productivity.

When task packages are prioritized, the space interference of target objects should be considered in order to improve the productivity of field operation. The task package to be scheduled, which has no target object that has space interference with the target objects of task packages in progress, has higher priority. Even though a task package has no space interference, if the task package has a source that is the sibling work cell of a task object of task packages in progress, it should not be scheduled for the available equipment group to avoid possible space interference. If there are several task packages that have no space interference, then the next rule, Shortest Travel Distance First (STDF), is applied.

Rule 2: Shortest Travel Distance First (STDF)

If a task package should be scheduled for an available equipment group and there are several task packages that have the same priority on NSIF, the task package nearest to the source object of the previously completed task package should be scheduled first, in order to minimize the travel distance. If there are also several task packages that have the same priority on STDF, then the last rule, Random (RAND), is applied.

Rule 3: Random (RAND)

Remained ties can be broken by random order. Any task package is randomly scheduled for an available equipment group.

3.2 Experimental Study of Scheduling Heuristics

To schedule multiple task packages, heuristic rules for sequencing task packages are used to prioritize them. In order to measure the performance of scheduling, tests of the TPSR rule against two other rules, FIFO[1] and Random[2], are carried out by simulation.

The simulation experiment is composed of 18 example cases: Case 1-1 through 2-3 have 16 work cells and 8 task packages, Case 3-1 through 4-3 have 48 work cells and 20 task packages, and Case 5-1 through 6-3 have 64 work cells and 30 task packages. Each case is provided with the distance of any two source objects ranging from 50m to 1000m in the given environment. It is assumed that total cut volume is equal to total fill volume in each case, and two or three groups of equipment are involved in field operations at the same time.

The simulation test bed is implemented on Intel Pentium II 366 computer, and is written in C++ language.

Upon completion of testing each case, two major performance values are measured for evaluation: (1) Space interference rate (SIR), and (2) Total travel distance (TTD). These criteria are closely related with the productivity and duration of the field operation project.

Space Interference Rate (SIR)

The SIR is the simple percentage of the number of task packages that have space interference over that of all task packages. It can be shown as follow:

$$SIR = \frac{TP_{SI}}{TP_{all}} \times 100 \qquad (1)$$

where, TP_{SI} = the number of task packages that have space interference, and TP_{all} = the number of all task packages in the activated list.

The minimization of the SIR is used as a primary performance criterion, because it reflects the efficiency of the scheduling heuristic rules. The larger SIR of experiments means that the applied heuristic rule generates the longer schedule for field operations due to high space interference that will result in low productivity.

Total Travel Distance (TTD)

The travel distance (TD) is a measure of the distance between the source of the task package most recently scheduled for an equipment group, and the source of the task package to be scheduled for the same equipment group.

$$TTD = \sum TD \qquad (2)$$

The shorter TTD indicates the field operations can be completed more quickly, because after completing a given task package, each equipment group can move to the next nearest work cell to start a new task package as early as possible. If a heuristic rule generates the schedule for each equipment group ineffectively, the travel distance is increased, which will result in the longer project duration.

3.3 Experimental Results

The average SIR for two and three equipment groups are presented in Figures 8 and 9 in the form of bar charts. As seen from the figure, as the number of task packages is increased, the SIR of RANDOM and TPSR is decreased and converges at zero percentage. However, the SIR of FIFO is not decreased, because task packages in the activated list are sequentially scheduled for equipment groups, which have a lot of chance to create space interference. One thing to note here is that if the initial number of task packages in the

[1] Task packages are sequentially scheduled by the order of entry into the activated task package list.
[2] Task packages in the activated list are randomly scheduled.

activated list is small like Case 1-1 through Case 2-3, only one or two task packages that have space interference can produce a high percentage of SIR. In spite of this fact, TPSR produces relatively low percentage of SIR with small number of task packages. To sum up the analysis, it is found that TPSR performs best, on average, among the three heuristic rules.

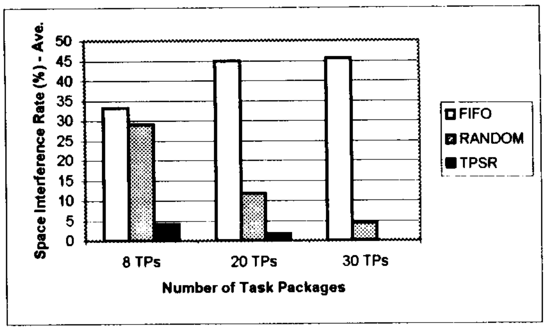

Figure 8. Average SIR for Two Equipment Groups

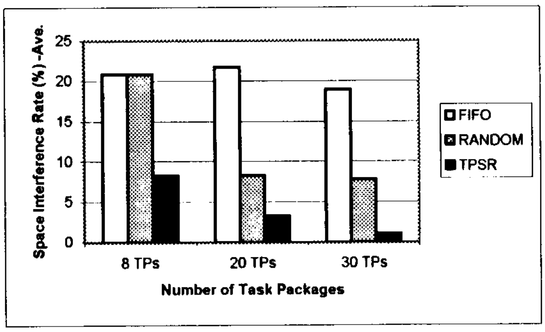

Figure 9. Average SIR for Three Equipment Groups

As shown in Figures 10 and 11, the TTD of all heuristic rules is seen to increase with a larger number of task packages. One particular thing found here is that, in the case of FIFO and RANDOM, the increase rate of TTD is very steep, however, that of TPSR's TTD is gentle. Across the various cases, the TPSR results in TTD values that are on average lower than FIFO by 36% and RANDOM by 50%, and can generate an effective schedule in terms of travel distance.

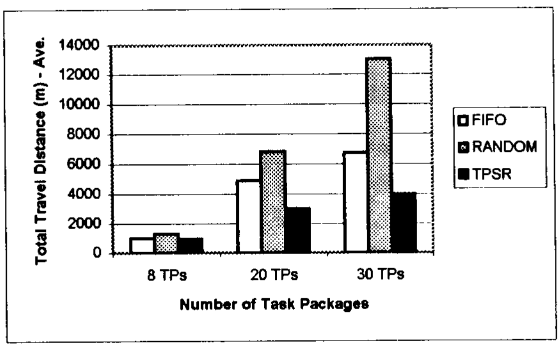

Figure 10. Average TTD for Two Equipment Groups

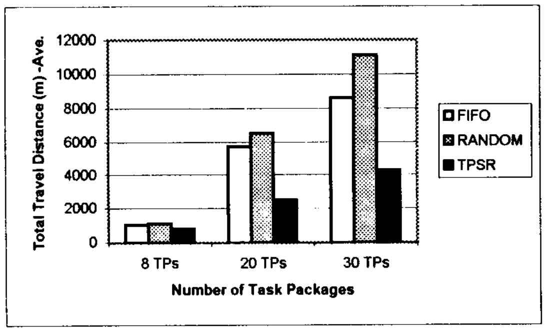

Figure 11. Average TTD for Three Equipment Groups

4. SUMMARY

This paper has presented the task identification and planning methodology. The construction site is partitioned into several work cells using the Quadtree algorithm. Information on the location of work cells, cut and fill volume, characteristics of soil, etc., are stored in quadtree data structure. Task objects are constructed to describe the characteristics of filed operation tasks to be completed in a certain work cell. A set of correlated task objects constructs a task package which includes information on how much material volume is transported to where. To identify a source work cell and target work cells of a task package, the source and target object selection algorithm is presented. To generate the schedule of task packages for each equipment group, a heuristic rule, Task Package Sequencing Rule (TPSR), is developed. This research conducts 18 experimental tests and compares the performance of the three scheduling heuristic rules (FIFO, RANDOM, and TPSR) based on two major evaluation criteria: (1) space interference rate (SIR) and (2) total travel distance (TTD).

Based on the above experimental test results, it is found that the developed heuristic rules, TPSR, performs better than the competing heuristic rules such as FIFO and RANDOM on performance measures, SIR and TTD. Even though the experimental tests are limited, the consistent results give a strong indication of the performance of task package scheduling.

5. REFERENCES

[1] Kim, S. K. and Russell, J. S. "Framework for An Intelligent Field Operation System (IFOS): Part 1 – System Architecture." *18th International Symposium on Automation and Robotics in Construction*, Krakow, Poland, 2001.

[2] Subramaniam, V., Ramesh, T., Lee, G., Wong, Y., and Hong, G. "Job Shop Scheduling with Dynamic Fuzzy Selection of Dispatching Rules." *International Journal of Advanced Manufacturing Technology*, Vol. 16(10), 759-764, 2000.