NOTAS DE MATEMATICA

№ 138

HEURISTIC ALGORITHMS
FORMAL APPROACH BASED ON COMPACT KNOWLEDGE-BASED MODEL

BY

EWA DUDEK-DYDUCH

PRE-PRINT

UNIVERSIDAD DE LOS ANDES
FACULTAD DE CIENCIAS
DEPARTAMENTO DE MATEMATICA
MERIDA-VENEZUELA
1993

HEURISTIC ALGORITHMS
- FORMAL APPROACH BASED ON COMPACT KNOWLEDGE-BASED MODEL

Ewa Dudek-Dyduch
Institute of Automatics,
University of Mining and Metallurgy
al. Mickiewicza 30, 30-059 Kraków, Poland
tel\fax: + 48 12 34 15 68

KEY WORDS: heuristic algorithms, greedy algorithms, local optimization, approximate branch & bound algorithms.

ABSTRACT

The paper deals with approximate algorithms that use a model ofmultistage decision process. The formal definition of multistage processes is presented as a specialized form of knowledge-based model. Then the three stage conception of synthesis of approximate algorithms that use local optimization is given. The local task is formulated as a multicriteria one and is based on measuring distances in subspaces of states. An example that illustrate the presented conception is given. Then the heuristic algorithms based on branch & bound method are discussed. Both class of considered algorithms are appropriated for many tasks of combinatorial optimization.

1. INTRODUCTION

The paper deals with a special class of heuristic algorithms worked out for optimization of so-called discrete determinable event processes (DDEP) defined in the author's previous paper. A large class of combinatorial problems corresponds to optimization of DDEP.

As it is known, heuristic algorithms are based on intuition, experience, observation but they have no sufficiently convincing elucidation and do not

guarantee that the optimal solution will be found. Very often an optimization problem as well as its algorithm are presented by means of verbal description only. It makes comparison different algorithms conceptions very difficult or even impossible. Recently these algorithms have become a subject of intensive studies. The studies relate to:

- classification and formalization of the algorithms,
- determining some properties of the algorithms such as convergency, computational complexity, error estimation etc.

The research are carried out mainly in the field of artificial intelligence and operational research. Especially, within the frame of artificial intelligence, one attempts to formal elucidation of heuristic algorithms ideas and giving some rules for creating them. The paper is connected with this direction of research. It presents formal approach to analysis and synthesis of some class of heuristic algorithms. The basis for the approach is assumption that heuristic algorithms can be considered as exact ones but solving simplified problems or as the algorithms realizing simplified way of problem solving. In order to indicate the introduced simplification it is necessary to present both problems and algorithms by means of common formalization.

There are three main general formal models used to solve combinatorial tasks: discrete (mixed) programming model, knowledge based models and state graph model. The paper deals with the class of heuristic algorithms that can be analysed with use of compact knowledge-based model devised by the author.

2. MULTISTAGE PROCESS AS A SPECIALIZED FORM OF KNOWLEDGE-BASED MODEL

Let us recall some notions introduced in the previous papers.

Multistage decision (constructing) process is a process P defined by the following six (U, Y, y_o, f, Y_N, Y_F) where the individual values are defined as follows: U is a decision set, Y is set of states, y_o, Y_N, Y_F are distinguished initial state and sets of not admissible and final states, respectively, f is a partial function

$f : U \times Y \to Y$ defined by means of the function $g : Y \to 2^U$ in the following way: $(u,y) \in \mathrm{Dom}\, f \Leftrightarrow u \in g(y)$. Thus function $g(y)$ determines the decision subset $U_p(y)$, for which function f is determined in the state y.

Function f was defined as a partial function. Thanks to it, all limita-

tions concerning the decisions can be taken into account with use of definition of the so-called sets of possible decisions in state y. The sets are denoted as $U_p(y)$ and defined as:

$$U_p(y) = \{ u \in U : (u,y) \in \mathrm{Dom}\ f \}.$$

At the same time an individual process P is represented by a set of its trajectories. A trajectory that ends in the set of final states is an admissible one and the decision sequence determining the admissible trajectory is an admissible decision sequence.

The task of optimization lies in finding of such an admissible decision sequence $\tilde{u}$ which minimizes a certain criterion Q. Thus an optimization task is determined by the pair (P,Q) where process P represents all the task limitations. A defined optimization problem is such a set of optimization tasks which have a commonly determined set of data (the same type of data fulfilling the commonly determined properties), common (parametric) definition of set of multistage processes and common (parametric) definition of criterion functions.
A lot of combinatorial tasks (problems) can be formulated in terms of this model (optimization of discrete manufacturing processes, scheduling problems, traveling salesperson problem, as well as classical tasks considered within the frame of artificial intelligence e.g. 8-Puzzle or block world problem). In this paper we will consider only the tasks and problems defined above. They will be named the tasks (problems) of composite object searching.

In the most general case sets U and Y may be presented as a cartesian product $U = U^1 \times U^2 \times \ldots U^m$, $Y = Y^1 \times Y^2 \times \ldots Y^n$.

Thus, the decision u as well as the state y is represented by finite sequence of values that belong to succeeding sets. We will use the notions "state vector" and "control vector" by analogy to the classical control theory:

$$u = (u^1,u^2,\ldots u^m) \qquad x = (x^1,x^2,\ldots x^n)$$

There are no limitations imposed on the sets, in particular they need not be numerical ones. Thus values of particular coordinates of state may be names of elements as well as some objects (e.g. finite set, sequence etc). Particular coordinates of the vector $u = (u^1,u^2,\ldots u^m)$ represents a separate decisions. This paper is appropriated for the processes for which at least one set

Y^i is not numerical one. The sets Y_N, Y_F and U_p are defined by means of logical formulae therefore the model is a special kind of knowledge based model.

The presented model is more general than the state graph one as the set of decisions need not be countable one. On the other hand it enable us to define more properties than the graph (e.g. one can analyse a structure of a state and decision as well as changes of particular coordinates).

Example 1.

Let us give the multistage decision process for the knapsack problem. The problem is as follows: given the set of objects $\mathbb{Z}$, the function $\tau : \mathbb{Z} \to \mathbb{R}^+$ that determines the weight of each object, the function $w : \mathbb{Z} \to \mathbb{R}^+$ that determines each object value and maximal weight d.. Which objects should be put into the knapsack so that the total their weight be less than d and the total value be maximal?

A state of the process is determined by the subset of objects in the knapsack. Thus the set of state $Y = 2^{\mathbb{Z}}$.

A decision lies in choosing the object to be put into the knapsack. Thus the set of decision $U = \mathbb{Z}$.

The initial state $y_o = \emptyset$.

The set of not admissible states

$$Y_N = \{y \in Y : \sum_{z \in y} \tau(z) > d\}$$ (all subsets of objects for which the total weight is grater than the maximal one.)

The set of final states $Y_F = Y \setminus Y_N$.

Sets of possible decisions are defined: $U_p(y) = \mathbb{Z} \setminus y$,

$U_p(y_{i+1}) = U_p(y_i) \setminus \{u_i\}$.

The function f is defined: $f(u,y) = y \cup \{u\}$.

Let us notice that value of the state coordinate is a set.

Others examples of combinatorial tasks model led by the multistage process are given in [4].

3. ALGORITHMS BASED ON LOCAL OPTIMIZATION

The most popular heuristic algorithms are those based on local optimization. They used the specially created function or local optimization task for choice of the "best" decision at each state of the generated trajectory.

Algorithms of this type are sometimes named "greedy" algorithms. Note that there are no formal definition of "greedy" algorithms. The function for local optimization is called a preference function or simply heuristics. Heuristics are created by means of intuition and very often more than one heuristics is proposed for the same problem. They are verified by means of computer experiments or sometimes by means of error estimation. The questions arise:

- can we create preference function or local optimization task not intuitively but consciously and, if yes, in which way?
- can we compare formally different algorithms of this type and indicate the introduced simplification?

The further considerations relates to the answer this questions.

The considered optimization task (P,Q) is assumed to be such that:

1. the criterion Q is additively separable,
2. process P is finite (i.e. its trajectories are finite),
3. each trajectory of the process P can be generated by means of polynomial algorithm.

Note that most of the known discrete manufacturing control optimization problems satisfy this assumptions.

Now we will present certain general method for designing the heuristic algorithms of this type. The method consists of the three stages.

Stage 1

At this stage one formulates a new, usually simplified task. As a result of verbal description of the problem, some necessary and (or) sufficient conditions for the optimal solution can be formulated. Usually, a simplification of the task is introduced at the same time. It lies in omitting some part of conditions. The reasons for the simplification are as follows:

- it may be difficult to define all the conditions,
- the conditions may have very complicated shape (e.g. it may be alternative or conjunction of many conditions),
- it may be difficult to find the trajectory for which the conditions are fulfilled.

Note that this stage is pretty similar to the one existing in the optimization of analytical problems.

In particular, we can distinguish the following types of conditions:

- conditions relating to direct decision influence on criterion,
- conditions relating to states i.e. determining the state sets that are advantageous (or disadvantageous) from the criterion point of view or due to possibility of generating an admissible trajectory;preference coefficients can be also determined for the sets, that take into account different consequences of running through the sets,
- conditions defining some indirect aims, particularly the ones defining the state sets to be achieved in shortest time or under minimal value of a special auxiliary criterion.

Stage 2:

At this stage one determines local optimization task. The task is usually defined as a multicriteria one.

In order to determine the "possible best" trajectory, one must choose a proper decision $u \in U_p(y)$ at every new generated state y. The question is: in which way it can be done in the situation when one can know only the nearest consequences of the decision, i.e. when information about the direct successors of the considered state is available only?
One can use information about the distinguished at the first stage "advantageous" or "disadvantageous" states. Obviously one also uses information about the set of goal states Y_G, set of not admissible states Y_N and information about the sets of possible decisions. As we need the generated trajectory to run only through the advantageous states and to avoid the disadvantageous ones, it seems most natural to introduce any "measure of distance" in the state space, and to assume some local criterions. Maximizations (minimizations) of the criterions correspond to maximizations (minimization) of "distances" to the particular distinguished sets of states.

Taking into account the local change of the global criterion and maximization (minimization) of the mentioned distances, we obtain the substitute local problem. This new problem is usually a multicriteria one.

The next question is: what type of "measure of distance" can we applied? According to the author, a proper semimetric can be used as an approximate "measure of distance". Let us recall the difference between the notion of metric and semimetric. Metric ρ in a space X is a function $\rho : X \times X \to \mathbb{R}$ such that the following hold:

1) $\rho(x,y) = 0 \Leftrightarrow x = y$

2) $\rho(x,y) = \rho(y,x)$

3) $\rho(x,z) \leq \rho(x,y) + \rho(y,z)$

It results from these conditions, that the metric has a non-negative value for each elements of the space X. The first condition need not be true for a semimetric.

Note that any metric defined in a subspace ,Y' of a space Y is a semimetric in the space Y. Roughly speaking, when we use semimetric we measure the distance in corresponding subspace, thus we use only some state coordinates. As a consequence a "distance" between two states are determined on a basis of merely part of information coded in the state. We can distinguished two basic group of semimetrics:

- semimetrics using numerical coordinates of state,
- semimetrics using coordinates of a higher order i.e. such coordinates values of which are sets and relations given explicitly (definition of order of objects is given in [8]).

The most simple semimetric of the first type is

$$\rho_1(y,y') = |y^j - y'^j|$$

where y^j, y'^j are numerical values of j-th coordinates of the states y and y' respectively. More complicated semimetrics can be obtained when grater number of numerical coordinates are used e.g.

$$\rho_2(y,y') = \sum_{j\in J} |y^j - y'^j|$$

where J is some set of indexes of the numerical coordinates.

In general, we can base on Holder metrics

$$r = \left(\sum_{j\in J} |y^j - y'^j|^p \right)^{1/p} \quad \text{for } p = 2,3,..$$

An example of the second type semimetric is

$$\rho_3(y,y') = |\bar{y}^j - \bar{y}'^j|$$

where y^j, y'^j are not numerical coordinates and such that their values are finite sets (relations) and $\bar{y}^j$ denotes the power of the set.

A distance between a state and a state set Y' is usually denoted by the same symbol as the corresponding semimetric ;

$$\rho(y,Y') = \min \{\rho(y,y') : y' \in Y'\}.$$

Stage 3

At this stage one should determine the manner of solving the local optimization task. Let us notice that the task has no analytical model, another words its solution is a certain decision that belongs to the set U (a value of decision need not be an element of the numerical space). As a consequence, one can not utilize directly methods of multicriteria optimization that use such notion as derivative, direction and gradient. Nevertheless, the basic idea of multicriteria approach can be applied here. Basing on the [11] one can characterize the main solving manners as follows:

a) arbitrary choice of a decision from the Pareto set; obviously, this manner is sensible in case when in each state y the Pareto set is much less than the set of possible decision $U_p(y)$,

b) assuming some weight coefficient (priorities) for the criteria and lexicographical choice,

c) majority choice that consists in the choice of such u^* for which the greater part of criterions have betters values,

d) choice that uses socialization lying in ascribing weights coefficients to the particular criterions

Thus the final type of the algorithm depends on the manner in which the local optimization task is solved.

Example 2.

Let us consider the well known greedy algorithm for the knapsack problem. The algorithm consists in ordering the objects according to non-decreasing values of $w(j)/\tau(j)$. The solution is the longest admissible sequence of objects that is consistent with this ordering.

Let us present the elucidation of the algorithm, based on the multicriteria approach. Let t_i denotes the total weight of objects that are in the knapsack in the state y_i ,

$$t_i = \sum_{z \in y_i} \tau(z)$$

(t can be considered as an output coordinate) and let $U_d(y)$ denotes the set of admissible decision in the state y i.e.

$$U_d(y) = \{u \in U_p(y) : f(u,y) \notin Y_N\}.$$

Criterion $Q = \sum_{i=0}^{n} \Delta Q_i$ where ΔQ_i denotes the increment of the criterion and n denotes the trajectory length. As the criterion is additively separable and monotonously increasing along the trajectories of the process [1,3], the value of it depends on a local increment ΔQ as well as on the trajectory length n.

As the set of the not admissible states is of the form:

$$Y_N = \{y \in Y : \sum_{z \in y} \tau(z) > d\}$$

and coordinate t is monotonously increasing and $\Delta t = t_{i+1} - t_i$ has finite value, thus the local optimization task consists in the choice of such a decision that maximizes ΔQ and at the same time keeps the maximal distance to the set Y_N.

Assuming the semimetric

$\rho(y,y') = |t - t'|$ and $\rho(y,Y_N) = |t - d|$

we obtain the local bicriteria task of the form:

$Q = (q_1, q_2)$,

$$\max q_1 = \max_{u \in U_d(y_i)} \Delta Q_i = \max_{u \in U_d(y_i)} w(u)$$

$$\max q_2 = \max_{u \in U_d(y_i)} \rho(f(u,y_i), Y_N) =$$

$$= \rho(y_i, Y_N) - \min_{u \in U_d(y_i)} \tau(u) = |t_i - d| - \min_{u \in U_d(y_i)} \tau(u).$$

Consider the following onecriteria task:

$$\max q = \max_{u \in U_d(y_i)} w(u)/\tau(u)$$

It is easy to see that any solution of this task belongs to the Pareto set of the previous bicriteria task.

4. HEURISTIC ALGORITHMS BASED ON B&B METHOD

Branch and Bound (B&B) algorithms to solve problems represented by the defined above multistage process model are analyzed in the author's paper [3]. Now, let us discuss the heuristic algorithms that are based on B&B method.

Generally speaking, B&B method lies in the constructing a decision tree, the nodes of which correspond to the sets of admissible solutions. By way of

eliminating the nodes, the successors of which do not contain admissible or better than the best solution, the number of calculations is reduced. The B&B algorithms have very valuable property. Namely, if we stop computing before the optimal solution is found then we obtain approximate solution criterion value of which can be estimated by means of lower bound value. We can improve the approximate solution when continue the computation. It is possible, however, only when algorithm of a single iteration is a polynomial one. If it is not polynomial then some simplification must be done, what causes that that B&B algorithm becomes a heuristic one. We can distinguished some kind of B&B based heuristic algorithms depending on:

- conception of lower bound creating,
- conception of upper bound creating,
- selection rule applied to the choice of next node for branching,
- elimination rules.

Let $L(\omega)$ stands for lower bound for a node ω and $Q(\omega)$ stands for minimal value of criterion function for set of solutions corresponding to the node ω. The criterion is assumed to be minimized.

As it is known, an algorithm is exact (when no implementation restrictions are assumed) only if the relation $L(\omega) \leq Q(\omega)$ holds for each node ω. It can be, however, very difficult or even impossible to create a lower bound that is effective and at the same time is easy to calculate (in particular, such one that can be calculated in polynomial time). Then, as a result of compromise, one can propose lower bound for which the above relation does not hold for all the nodes but it holds for possibly great number of the nodes. It deals also with the lower bound obtained by means of relaxation (see [3]). Then two cases can occur: the relaxed problem is polynomial one or it is NP-hard. In the last case we can obtain the value of lower bound only as approximate one. Certain probabilistic analysis such a case is presented in [10].

If the problem of admissible solution search is NP-hard then it can be difficult to find any admissible solution in reasonable time. We can use the approximate value of upper bound then. If the approximate value is too large then no subset of solutions can be eliminated. If it is too small then all solutions can be eliminated.

Let us consider constructing algorithms [3]. The selection rules can utilize the values of lower bound (exact or approximate one) or can be based on local optimization tasks defined in the previous section.

Heuristic elimination rules can be also based on "distance" to the set of not admissible states Y_N, that is expressed by means of any semimetric.

It must be point out that heuristic lower bound, upper bound as well as heuristic elimination rules influence the solution (i.e. optimal solution can be never found). On contrary, heuristic selection rules influence only the time of solution search (when there are no restriction referring to a computer memory or time).

Let us also notice that if the lower bound of the first iteration is correct then an error of heuristic algorithm can be evaluated. This evaluation, however, can be very rough.

5. CONCLUSION

The paper presents the formal approach to analysis and synthesis of heuristic algorithms of certain type. The algorithms are appropriated for many combinatorial tasks and problems that can be modelled with use of general knowledge-based model.

The 3-stage conception of synthesis of heuristic algorithms that use local optimization is given. The local task is formulated as a multicriteria one and is based on measuring distances in subspaces of states. The approach is generalization of the one presented in [10], that uses a model of state space graph. The elucidations given there are not general but are based on intuition. Moreover, they are presented by means of examples for which local optimization tasks are one criteria ones. The introduced in this paper formalism enable us to discuss these algorithms in more detail.

The second type of heuristic algorithms considered in the paper are those based on branch and bound method. Different simplification are discussed.

The presented formal approach enable us to compare different heuristic algorithms (based on the same type of model) and discuss the introduced simplifications. In the similar way one can analysed the heuristic algorithms based on the other optimization methods e.g. heuristic algorithms based on decomposition method or the ones based on dynamic programming.

REFERENCES

[1] E. Dudek-Dyduch, Formalization and analysis of problems of discrete manu-

facturing processes (in Polish), Zesz. Nauk. AGH, z.54,1990.

[2] E. Dudek-Dyduch, Formal bases of Classification of Discrete Production Processes Control Problems, Zesz. Nauk. AGH Nr 1313, Automatyka z.49, 8, 1989, pp 189-200.

[3] E. Dudek-Dyduch, Control of discrete event processes - branch and bound method. Prepr. of IFAC/Ifors/Imacs Symposium Large Scale Systems: Theory and Applications, Chinese Association of Automation, vol. 2, 1992, pp 573- 578.

[4] Dudek-Dyduch E., Dyduch T.: Zastosowanie wieloetapowych procesów w optymalizacji sterowania dyskretnymi procesami produkcyjnymi, Zesz. Nauk. AGH s. Automatyka, z. 61, nr. 1492, 1992.

[5] Dudek-Dyduch E.: Discrete manufacturing processes - General knowledge based model, Proc. of III Intern. Workshop on Qualitative Reasoning and Decision Technologies Quardet'93 (accepted for publication)

[6] V. Kumar, L. Kanal, A general branch and bound formulation for understanding and synthezing and/or tree search procedures, Artifical Intelligence 21 (1,2) 1983, pp 178-198.

[7] Ligeza A. (1992) Logical foundations for knowledge-based control systems, Raport LAAS No 92269.

[8] W. Marciszewski i in., Mała encyklopedia logiki, Wyd. Ossolineum, Wrocław 1988.

[9] Nilsson N.J. (1980) Principles of Artificial Intelligence. Tioga Publishing Co., Palo Alto, California.

[10] J.Pearl: Heuristics. Intelligent search strategies for computer problem solving. Addison-Wesley Comp. Menlo Park, 1984

[11] L.A. Szołomow, Łogiczeskije mietody issledowanija diskretnych modielej wybora, Izdat. Nauka, Moskwa 1989.