DEPARTMENT OF NATIONAL DEFENCE
CANADA

Operational Research • Recherche opérationnelle
A Posse Ad Esse

OPERATIONAL RESEARCH DIVISION

DIRECTORATE OF OPERATIONAL RESEARCH (JOINT & LAND)

DOR(J&L) RESEARCH NOTE RN 9824

GOAL PROGRAMMING FOR STRATEGIC PLANNING

BY

Mr. R. Dickson

DECEMBER 1998

OTTAWA, CANADA

National Defence Défense nationale

OPERATIONAL RESEARCH DIVISION

CATEGORIES OF PUBLICATION

ORD Reports are the most authoritative and most carefully considered publications of the DGOR scientific community. They normally embody the results of major research activities or are significant works of lasting value or provide a comprehensive view on major defence research initiatives. ORD Reports are approved personally by DGOR, and are subject to peer review.

ORD Project Reports record the analysis and results of studies conducted for specific sponsors. This Category is the main vehicle to report completed research to the sponsors and may also describe a significant milestone in ongoing work. They are approved by DGOR and are subject to peer review. They are released initially to sponsors and may, with sponsor approval, be released to other agencies having an interest in the material.

Directorate Research Notes are issued by directorates. They are intended to outline, develop or document proposals, ideas, analysis or models which do not warrant more formal publication. They may record development work done in support of sponsored projects which could be applied elsewhere in the future. As such they help serve as the corporate scientific memory of the directorates.

ORD Journal Reprints provide readily available copies of articles published with DGOR approval, by OR researchers in learned journals, open technical publications, proceedings, etc.

ORD Contractor Reports document research done under contract of DGOR agencies by industrial concerns, universities, consultants, other government departments or agencies, etc. The scientific content is the responsibility of the originator but has been reviewed by the scientific authority for the contract and approved for release by DGOR.

DEPARTMENT OF NATIONAL DEFENCE

CANADA

OPERATIONAL RESEARCH DIVISION

DIRECTORATE OF OPERATIONAL RESEARCH (JOINT & LAND)

DOR(J&L) RESEARCH NOTE RN 9824

GOAL PROGRAMMING FOR STRATEGIC PLANNING

by

Mr. R. Dickson

Recommended by: G. L. Christopher
SPORT

Approved by: G. Lafond
DOR(J&L)

Directorate Research Notes are written to document material which does not warrant or require more formal publication. The contents do not necessarily reflect the view of ORD or the Canadian Department of National Defence.

OTTAWA, ONTARIO

DECEMBER 1998

ABSTRACT

This research note outlines how a Goal Programming methodology could be used as an analysis tool for scenario-based strategic planning of the Canadian Forces. A sample problem is put forward and solved in a Linear Programming framework. Implementation details and computer code for the algorithm are provided in Annex sections of this note.

RÉSUMÉ

Cette note de service décrit comment la méthode de programmation par objectifs pourrait être utilisé pour l'analyze de planification stratégique par scénario pour les forces armées canadiennes. Un problème démonstratif est décrit et résous utilisant la programmation linéaire. L'implantation et le code de l'algorithme sont fournis dans une annexe.

TABLE OF CONTENTS

PAGE

LIST OF ILLUSTRATIONS

ACKNOWLEDGEMENTS

The author would like to acknowledge the contributions of his colleagues within the Strategic Planning Operational Research Team in developing the ideas put forth in this paper. In particular, observations, suggestions and criticisms from P. Comeau, and S. Isbrandt were most helpful and appreciated.

GOAL PROGRAMMING FOR STRATEGIC PLANNING

Introduction

1. Within the Directorate of Defence Analysis (DDA), a project is currently underway to develop a means of helping the Department conduct long-term strategic planning for the Canadian Forces (CF). The Strategic Planning Operations Research Team's (SPORT's) primary role in this undertaking is the development of a methodology, or set of methodologies, to provide analytic support to the project. Several different methodologies are being considered for this purpose, and this research note is intended to outline one such possibility. In doing so, it must be noted from the onset that the methodology described herein will not provide "the" answer to the long-term strategic planning issue. Rather, it is envisioned as a tool that might be used as "part" of the analysis portion of the project.

Strategic Planning for the Canadian Forces

2. DDA has been tasked to develop a means of helping the Department perform long-term Strategic Planning. Toward this end, the group has been putting together a series of eleven Force Planning Scenarios (Ref. 1) with the input of representatives from across the Department. The scenarios have been selected to reflect the full set of tasks that the CF may be called upon to perform over the entire spectrum of conflict from peacetime operations to full combat.

3. Initially, the DDA-led group developing the scenarios put together short (2-3 page) snapshots of what each scenario would entail. At present, work is underway to expand upon these ideas to the point where they can support a detailed analysis of future force requirements. One of the requirements of the project that is being reflected in this

expansion is the identification of the full set of capabilities required by the CF. The premise of SPORT's involvement is to develop a means to assist in identifying the "best" capability mix that meets the needs of the CF across the full spectrum of conflict. This in turn can be used to drive the programs that acquire new equipment and maintain that on hand.

4. It should be pointed out that work has already been done by the various military services (environments) toward analyzing their own force structure packages with a view to the future. The work being done within DDA is not intended to replace any of these efforts. Rather, it is meant to consider the issue from a joint perspective to assist in force planning for the CF as a whole.

Goal Programming

5. Many different methodologies are presently being considered to assist in the strategic planning project. The one that will be outlined here is Goal Programming (Refs. 2,3). There are a number of different ways to implement such a routine, Dynamic Programming and Linear Programming being just two of the possibilities. Dynamic Programming as a candidate methodology has been outlined in Ref. 4. As such, we will present the Goal Programming methodology in a Linear Programming framework.

6. The basic principle underlying the Goal Programming methodology involves the specification of a goal or series of goals to be achieved, and then determining how closely it/they can be attained in a perscribed solution space, subject to a series of constraints. To illustrate this concept, consider an example in which there are three countable entities of interest. With each of these entities, we associate a desired quantity (the goal) denoted by G_1, G_2, and G_3 respectively. Also identified are the unit cost C_i for the items, as well as a set of values $\{X_i\}$, $i = 1,2,3$ where each X_i is itself a set of quantity levels that entity i can assume. The notation Q_i will be used to denote a particular level within the set X_i. The

problem is to determine the quantity levels that can be achieved to best meet the goals, while staying within a budgetary limit of B units.

7. Variables d_i, $i = 1,2,3$ are introduced to measure any deviations from the goal levels, i.e.,

$$d_i = G_i - Q_i \text{ , } i = 1,2,3.$$

8. Having identified the deviations from the target values, it is also important in the methodology to be able to distinguish between deviations the underachieve the goals and those that overachieve them. Toward this end, variables d_i^- and d_i^+, $i = 1,2,3$ are defined as:

$$d_i^+ = \begin{cases} 0, & \text{if } d_i >= 0 \\ -d_i, & \text{if } d_i < 0 \end{cases}$$

$$d_i^- = \begin{cases} d_i, & \text{if } d_i >= 0 \\ 0, & \text{if } d_i < 0 \end{cases}$$

which means $d_i = d_i^- - d_i^+$.

9. The ideal solution to the problem of course is to meet all the goals exactly. Should this not be possible, the best alternative solution would be to minimize the total deviations from the target levels across all of the goals. A Linear Programming formulation of the problem could be constructed as follows:

$$\begin{array}{lll} \text{Minimize}: & \sum_i (d_i^+ + d_i^-) & \\ \text{Subject to}: & Q_i + d_i^- - d_i^+ = G_i, & i = 1,2,3 \\ & d_i^+ \geq 0, & i = 1,2,3 \\ & d_i^- \geq 0, & i = 1,2,3 \\ & Q_i \geq \min(X_i), & i = 1,2,3 \\ & Q_i \leq \max(X_i), & i = 1,2,3 \\ & \sum_i Q_i C_i \leq B & i = 1,2,3 \end{array}$$

where $\min(X_i)$ and $\max(X_i)$ denote the minimum and maximum values in the sets X_i, i = 1,2,3 respectively. Note that the C_i are constants, so the final constraint in the model is indeed linear.

10. Of course, not all of the goals in a given problem may be of equal "value." It may be desirable to emphasize that deviating from one of the goals may be less preferable than deviating from another. As such, it is not uncommon when developing Goal Programming algorithms to apply weight terms to the deviations to stress the importance of meeting a particular goal or goals. Furthermore, for a given entity, underachieving the goal may have a quite different effect than overachieving it. Thus, the use of weight terms can be further refined to reflect a preference of deviating from the goal in one direction over another. This being said, a more general formulation of the above problem would be:

$$
\begin{array}{lll}
\text{Minimize}: & \sum_i (w_i^+ d_i^+ + w_i^- d_i^-) & \\
\text{Subject to}: & Q_i + d_i^- - d_i^+ = G_i, & i = 1,2,3 \\
 & d_i^+ \geq 0, & i = 1,2,3 \\
 & d_i^- \geq 0, & i = 1,2,3 \\
 & Q_i \geq \min(X_i), & i = 1,2,3 \\
 & Q_i \leq \max(X_i), & i = 1,2,3 \\
 & \sum_i Q_i C_i \leq B & i = 1,2,3
\end{array}
$$

where the w_i^+ and w_i^- terms denote the weights associated with overachieving and underachieving the goals respectively.

11. The simplfied example put forward here does not do complete justice to the Goal Programming methodology as problems are typically much more complicated, particularly in terms of the constraints to be met. But the idea of defining non-negative variables to capture any over/underachievement of the desired goals, with a view toward minimizing the (perhaps weighted) sum of these variables is the basis of the methodology

and the work outlined in this research note. The interested reader is referred to Refs. 2 and 3 for further particulars of how the methodology works.

Goal Programming for Strategic Planning

12. We now turn our attention to applying the Goal Programming methodology to the strategic planning problem facing DDA. This section will describe in general terms how the methodology may be used, while the next section will look at a more specific, yet still rather general, example of the problem. To implement the methodology and example chosen, the software package AMPL Plus® (Ref. 5) has been used. For this particular implementation, two ASCII text files have been created for input into the software. One of the files contains the model, including the statement of the objective function and constraints, as well as the declaration of all variables and parameters used. It can be found in ANNEX A. AMPL Plus® requires this file to have a .mod file extension. The other file, located in ANNEX B with a .dat extension, displays all parameter values that are used by the model.

13. For the Force Planning Scenarios project, the goals will typically be associated with capability sets desired at given points of time in the future. How these desired sets are determined is beyond the scope of this paper and will not be discussed in detail here. Suffice to say that they may be determined via a strategic assessment or perhaps through expert opinion and consensus on the part the three environments. Another possibility would be to analyze the eleven scenarios for the probability of occurrence over specified future time periods and then determining desired capability mixes based on the resulting need.

14. The Goal Programming approach for the strategic planning problem begins with an initial capability mix state which could be viewed as the present make-up of the Canadian Forces. For each capability, a series of levels are determined and indexed by the non-negative integers (0,1,2,3,...). Although indexed linearly, these levels are not to

be considered as uniform stepwise increases in the capabilities, nor are the indices to be considered physically meaningful in and of themselves. For instance, levels 1-4 for a field hospital may correspond to a 100-, 300-, 500-, and 1000-patient facility respectively. Because of this, the quantitative values associated with the capability levels must also be maintained. We would like to be able to compare one level of a capability to another, but it would not be correct to say, in general, that a 300-patient field hospital has three times the "value" of a 100-patient facility. As such, a utility function must be used in the methodology to "measure" the functionality of the various levels and facilitate comparisons between the levels.

15. For each of a series of time periods, a desired mix of capabilities is selected and their corresponding utility is determined. For each of the capabilities, weight terms are specified to penalize any deviations from the desired mix. (Technically, negative reward weights could be given for any deviations in excess of the desired mix, but this leads to problems in the present formulation of the model. In that type of environment the model would attempt to overacquire the capabilities as much as it could in an effort to give the objective function to be minmized the largest negative value possible). Additional weight terms are also given for each of the time periods involved as one may wish to emphasize the importance of acquirng the desired capability mix at one point in time over another. The basic idea behind the methodology is to analyze the possible solution space determined from the defined capability levels, and find the capability mix that, for a given time period, minimizes the sum of weighted utility deviations from that of the desired capability mix, weighted once again over the time periods under consideration.

16. For a given capability, the various period-level combinations can be viewed as a network with the output of the model identifying a path between the selected nodes over time. Figure 1 below details a possible path followed for a capability with five levels over four time periods.

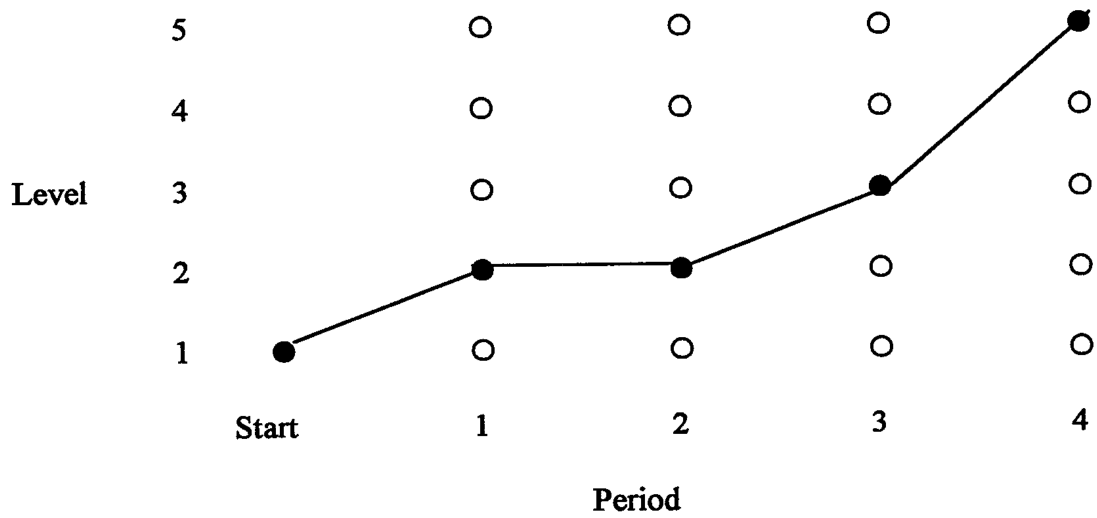

Figure 1: Sample Capability Level Path over Four Periods

17. For modelling purposes, a series of binary indicator variables are used to trace the path followed between the nodes over time (where a value of 1 indicates that a path between nodes is taken and zero otherwise). Three constraints are used in the model to identify the path followed:

```
subject to Start {c in CAP, i in LEVEL[c]:i=StartLevel[c]}: sum{j in LEVEL[c]}
        I[c,i,j,1]=1;

subject to Start2 {c in CAP,i in LEVEL[c]:i<>StartLevel[c]}: sum{j in
        LEVEL[c]} I[c,i,j,1]=0;

subject to Balance {p in 2..TotalPeriods,c in CAP,i in LEVEL[c]}: sum{j in
        LEVEL[c]} I[c,i,j,p]=sum{k in LEVEL[c]} I[c,k,i,(p-1)];
```

The indicator variable I[c,i,j,p] is to be read as a variable for capability c, going from level i in period p-1 to level j in period p. The "Start" constraint states that for all capabilties c, the path followed begins at the inital level specified in the model parameter set. Note that the means employed to ensure that a single path is followed through the network is to require that the sum of the 0-1 indicator variables be equal to 1. The second constraint (Start2) ensures that the path followed does not begin at a level other than the specified start level. The third constraint is a means of ensuring that the level we arrive at in a time period subsequent to the starting period was reached from where we actually were in the previous time period.

18. Of the other constraints that the model is subject to, the most noteworthy are those pertaining to the budget:

```
subject to Start_Budget: B[1]=BB[1];

subject to Subsequent_Budget {p in 2..TotalPeriods}: B[p]=BB[p] + (B[(p-1)] -
        C[(p-1)])*INT[p];

subject to Spent {p in PERIOD}: C[p]=sum{c in CAP,i in LEVEL[c],j in
        LEVEL[c]} CC[c,i,j,p]*I[c,i,j,p];

subject to Cost_Constraint {p in PERIOD}: C[p] <= B[p].
```

The data requirement for implementing the model (ANNEX B) involves the cost of making transitions between the different levels of a given capability over the various time periods under consideration, as well as any maintenance costs that may be involved (the CC[c,i,j,p] variable). As well, a baseline budget, BB[p], for each of the time periods must be specified. This budget may be augmented in a given time period by any monies not spent in the previous period (with a multiplicative factor built in to accommodate any interest accrued, (INT[p])), but at no time may the total cost of transitioning the capabilities between levels exceed the monies in the budget. Note that in the "Spent" constraint which calculates the total monies spent in a given time period, we use the indicator variables I[c,i,j,p] as a form of "on and off" switch to flag those capabilities that are being acquired under the model. This switch will appear in a number of the other constraints as well.

19. As implemented, the algorithm accommodates capability transition costs that can change over time. While more realistic than a static costing routine, this does significantly increase the data requirement for the model. If need be, the algorithm is easily adapted to a static costing environment. Note further that while not used in the

sample problem presented here, negative costs are also allowed in the model. This could be used to accommodate any monies acquired through the selling of capabilities as levels are reduced.

20. Consistent with the introductory example used to outline the Goal Programming methodology, non-negativity constraints exist for the variables that measure any positive or negative deviations from the goal utility in each time period, as well as a demand constraint that actually calculates the deviations:

```
subject to Min_dm {c in CAP, p in PERIOD}: dm[c,p]>=0;

subject to Min_dp {c in CAP, p in PERIOD}: dp[c,p]>=0;

subject to Demand {c in CAP, p in PERIOD}: sum{i in LEVEL[c],j in
        LEVEL[c]}I[c,i,j,p]*Utility[c,j]+dm[c,p]-dp[c,p]=GSUtility[c,p]
```

where Utility[c,j] is the calculated utility of capability c at level j, and GSUtility[c,p] is the utility of capability c in Goal Set period p.

20. The other constraints in the model determine the minimum and maximum levels of a capability to consider acquiring in a given time period:

```
subject to Min_Obtain {c in CAP, p in PERIOD}: sum{i in LEVEL[c],j in
        LEVEL[c]} I[c,i,j,p]*j >= Cmin[c,p];

subject to Max_Obtain {c in CAP, p in PERIOD}: sum{i in LEVEL[c],j in
        LEVEL[c]} I[c,i,j,p]*j <= Cmax[c].
```

Goal Programming is done with the understanding that not all of the desired goals are likely be achieved. However, there may be a limit as to how far from the goals we may be willing to deviate, i.e., while it may be "acceptable" to underacquire a given capability by a certain amount, there may be a threshold beyond which we are not willing to entertain the possibilities. Part of the data requirement for the model involves the specification of these thresholds, and the difference between the goal levels and these thresholds determines the minimum level that is to be considered for the given capability. At the opposite extreme, the default level for the maximum capability level to consider is determined by the maximum level of the capability over the full spectrum of time periods. For example, consider the capability levels for the field hospital described above. In a four-period model, the goal levels for a hospital may be level 2 in periods 1 and 2, level 3 in period 3 and level 2 again in period 4. In this case, the maximum level to consider acquiring in any time period would be 3. One might be tempted to set the maximum level in a time period equal to the goal level, but this can lead to problems. For instance, in the example above, we would have no choice but would be "required" to reduce our level of field hospital capability in time period 4. This could be detrimental should it be desirable to have level 3 again in some subsequent time period.

21. Another important feature of the maximum capability level chosen involves costing. While cost is a constraint in the model and not part of the objective to be minimized, given the dynamic costing feature used, it may be less expensive to buy excess capability today with a view to the future. More on this will be discussed in a subsequent section.

22. As a final note on these maximum and minimum level constraints, once the Force Planning Scenarios project is fully developed, the corresponding solution space of possible capability levels is going to be huge. Even if only 100 capabilities are identified with 5 levels each, the solution space will consist of 5^{100} data points. These minimum and maximum levels could go a long way toward paring down the size of the space to consider in the model.

A Sample Problem

23. For illustrative purposes, a simplified version of the Force Planning Scenarios problem is put forward here. While pared down to an easily managed size, the

characteristics found therein are typical of those in the larger planning problem facing DDA.

24. As pointed out already, the data used in this sample problem can be found in ANNEX B. Three capabilities are considered in the example over four time periods. Each capability has three levels associated with it plus a zero level, and is considered to begin in time period 0 at level 1. The goal set quantities are representative of the desired capability levels to be achieved in each time period. For simplicity, these levels have been chosen to be non-decreasing over time. The case of desired capability levels that fluctuate up and down over time is of particular interest and will be discussed in a subsequent section.

25. The thresholds that determine how far below the desired capability levels to consider acquiring are of course small due to the simplicity of the sample problem. Note that some of the thresholds are given a value of 0. These represent hard constraints in the model where a desired capability level "must" be met in the given time period.

26. For simplicity, the weight terms for negative deviations from the goal have all been set to one. Weight terms for excess deviations have not been set in the model's parameter file, but have been assigned trivially small positive default values in the model itself. While it would be nice to be able to set these values to zero to indicate that the thought of acquiring excess capability does not bother us, this could cause difficulties depending on how the solver in one's Linear Programming package works because cost is not part of the objective function. For instance, if excess deviation weight terms have values of zero, then meeting the goal as well as exceeding it all contribute values of zero to the objective function to be minimized. Depending on how one's solver works, capability levels in excess of the goal may be chosen if they meet the budgetary constraints of the model. This could be done on an undesirable arbitrary basis and as such is best avoided by the means taken here.

27. Another way to deal with the problem that should be put forward for future consideration is a preemptive Goal Programming scheme. Here, one could set all the

excess deviation weights to zero, run the present model to determine all possible capability mixes that give the same value for the minimized objective function, and then feed these results into a second model that chooses the single option that minmizes cost.

28. Once again for simplicity, a static costing scheme has been chosen based on that used in Ref. 4. A constant baseline budget of 10 units has also been chosen, as well as a constant interest rate for any funds left over from one time period to another.

29. As for the utility function that is used to quantify the functionality of the given capability levels, there are no hard rules on the form they are to take. For this simplified example, the capabilities being considered (such as a field hospital) are of a type for which it is easy to envision that increasing the level leads to increased utility. Two forms of utility functions immediately stand out as possibilities to consider in such cases. One would be a form of exponential-type growth that quickly rises from a utility of zero for a capability level of zero, and then asymptotically tapers off to some threshold level as illustrated in Figure 2. Another would be a logistic growth-type function that also begins at a point (0,0) and then grows in an S-shape to a threshold level as in Figure 3. A simple function that covers both cases is $1-\exp(-\alpha(x/\max\{x\})^{\beta})$, where α and β are fixed postive constants, x is the quantitative value associated with a capability level, and max{x} is the maximum of these quantitative values across all time periods. (The two different forms are obtained by choices of β less than, and greater than 1 respectively). The max{x} term is of course optional as $(\max\{x\})^{-\beta}$ is a constant which could be incorporated into the α term, so we will work with the more simplified equivalent form of $1-\exp(-\alpha x^{\beta})$. For our sample problem, the parameters α and β have been chosen to be 1 for all capabilities, and the quantitative values of the capabilities have been taken to be the levels themselves, i.e., level 0 has a value of 0, level 1 has a value of 1, etc.

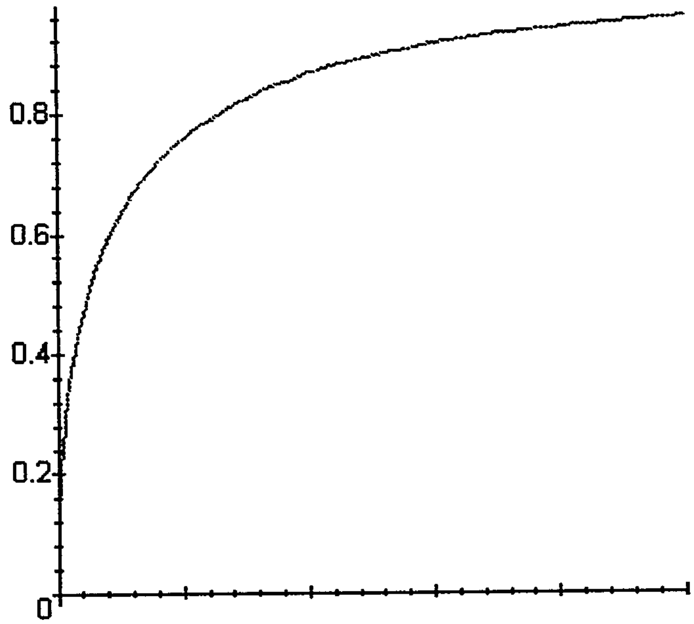

Figure 2. Exponential-Type Utility Function

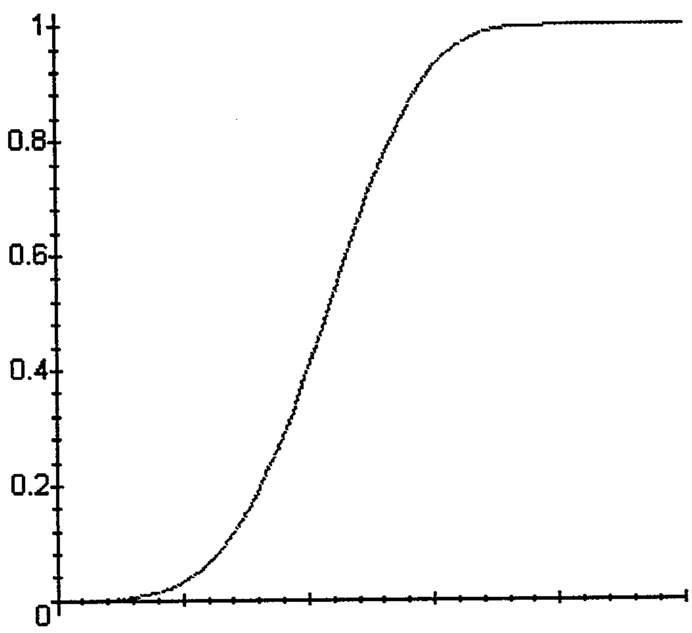

Figure 3. Logistic-Type Utility Function

30. Finally, a word should be said about the objective function used in the example. To facilitate easier reading, it has been broken down in the AMPL Plus® code in ANNEX A as an objective function and a constraint:

```
minimize Sum_Cap_Deviations: sum{p in PERIOD} pwt[p]*WD[p];

subject to Weighted_Deviations {p in PERIOD}: WD[p]=sum{c in
      CAP:GSUtility[c,p]>0} (wm[c,p]*dm[c,p]+
      wp[c,p]*dp[c,p])/GSUtility[c,p].
```

Instead of a straight linear weighted deviation measure, a percentage measure has been chosen, i.e., all deviations are weighed against the goal utility. This has been done because linear deviations have different meanings depending on the value of the goal. For example, if the goal utility was 0.99 on a 0-1 scale and the "best" affordable option had a utility of 0.98, the linear deviation would have a much smaller effect than if the goal utility was 0.02 and only 0.01 could be attained. In the second case, only 50% of the desired utility could be attained. This percentage scheme of course leads to problems if we wish to eliminate a capability (have a goal of 0) as a division by 0 situation would arise, but this is avoided by simply ignoring those terms in the sum to be minimized. This will not affect the final result as 0 levels are always attainable budgetwise if the transition has an associated cost that is non-positive.

31. A similar situation can arise if the model attempts to acquire a new capability, i.e., the case in which the goal levels are 0 for the first few time periods and positive from there onward. Should the model want to start acquiring some of the capability before the goals rise to a postive value, the same division by zero situation would present itself. Once again, those terms can simply be avoided in the objective function, provided that we truly mean that purchasing in excess of the goal is not disagreeable, nor rewardable. (Keep in mind that in setting the excesss deviation weights to a trivially small positve value, we really "mean" a value of zero. The token values assigned simply provide a means of differentiating the case of the goal being met to that of it being exceeded). Also keep in mind that in ignoring terms, we are only talking about the objective function

which, given the terms in question, will not be affected by the omissions. The remainder of the model will not be affected either and it will indeed keep track of the true path followed through the network of period-level combinations.

Sample Problem Results

32. Table I details the results of the implementation of the model with the data found in ANNEX B.

Table I

Sample Problem Results

Period		Goal Level (Acquired)					
	Period Weight	Cap 1	Cap 2	Cap 3	Budget	Cost	Weighted Deviation
0	n/a	1 (n/a)	1 (n/a)	1 (n/a)	n/a	n/a	n/a
1	1	3 (2)	1 (1)	1 (1)	10	4	0.0900306
2	1	3 (3)	1 (1)	1 (1)	16.6	5	0.0
3	2	3 (3)	2 (3)	1 (1)	22.76	20	0.0000989
4	4	3 (3)	3 (3)	1 (1)	13.036	12	0.0

The total weighted deviation measure weighted over the time periods is 0.0902284 based on the fact that the goals were not achieved in time periods 1 and 3 (0.0900306 + 2*0.0000989). Note that in both of these instances, the goals could have been achieved. In time period 1, the cost of taking capability 1 from level 1 to level 3 while maintaining capabilities 2 and 3 at level 1 would have been 9 units which is within the budgetary constraint of 10. In time period 3, the model actually chose to overacquire capability 2, obtaining a level of 3 instead of the goal of 2. In both cases, the apparent discrpeancy can be traced to the high penalty weight for not meeting the goal in time period 4. If all the goals are achieved in periods 1, 2, and 3, 18.811 units will be found in the budget for time period 4 whereas 22 units would be required to meet the needs of that period. If on the other hand, the goal of level 2 was met for capability 2 in time period 3 while the level of capability 1 rose from 1 to 2 in the transition from period 1 to 2, and then level 2 to 3 in period 3, 22 units would again be required in the budget in time period 4 yet only 21.836 would be available. Finally, the capability 1 goal of level 3 could be met in period 1 with level 3 being achieved for capability 2 in period 3, but this would leave 10.011 units in the budget for time period 4 whereas 12 units would be required for maintenance of the capability levels. Thus, the model has chosen the path that meets the most stringent need found in period 4 while minimizing the adverse effects that meeting this need has on the other time periods.

Observations

33. As was pointed out at the onset, Goal Programming will not provide "the" answer to the strategic planning problem facing DDA. In particular, it will not provide the answer as to what the "best" capability mix for the Canadian Forces will be. Rather, once a "best" or desired mix is "chosen", Goal Programming will be able to determine if that capability set is achievable, and more importantly, if not, what mix could/should be obtained to get as close to the goal as possible. But determining this optimal mix of capabilities is not without its challenges as will be discussed here.

34. Without a doubt, the biggest obstacle faced is the sizable amount of data required to implement the routine. The determination of the capability levels and the goal sets to be striven for has already been discussed and need no further comment to give an appreciation of the challenges to be overcome. But these challenges may very well pale in comparison to the determination of the "soft" parameters that the model requires, in particular the various weight terms necessary to implement the algorithm. What would a penalty weight of 2 for a field hospital actually mean? If it does indeed have a weight of 2, what weight does one assign to strategic lift? Such questions probably constitute a

complete study in themselves and as such will not be discussed in detail here, but they are concerns that must be addressed at some point in the project. How does the concept of risk come into play? Perhaps this provides an answer on the weights issue as they may be determined from the risk associated with not acquiring capabilities. If not, a post-Goal Programming analysis is almost certainly necessary to analyze the results against the desired mix, i.e., given that we "want" a particular set of capabilities at a certain time and the model determines the optimal solution to be something else, what is the risk involved in acquiring this something else instead?

35. The utility functions used in the model to facilitate capability level comparisons are also going to present a challenge to the project. It is going to be very difficult to quantify what each level means for each capability considered, and more difficult still to determine an analytic function to describe the complete quantification. Perhaps in the end, the model will have to be adapted to receive quantified utility inputs directly instead of using a function to compute the functionality.

36. These of course are difficulties that could be faced with any methodology used for the Force Planning Scenarios project. Of the challenges inherent to this methodolgy, the one that stands out is the concept of goal capability levels that fluctuate up and down over time. Suppose that for the field hospital example used throughout this paper, the desired levels in a four time period example are: level 2 in periods 1 and 3, level 4 in period 2, and level 3 in 4. A problem arises in that it might not be fiscally sound to reduce the level in the third time period only to buy it back again in period 4. (Note: it is difficult to envision a situation where the personnel involved in determining the goal levels would allow the oscillating fluctuation suggested to actually occur. It could happen however, at least in theory, if a software routine was used to generate the levels with the results being automatically fed into the Goal Programming algorithm). One way around this problem would be to add a costing term to the objective function to be minimized, but this leads to added difficulties as one must then determine a means of weighing costs against utility deviations. It could also be difficult to find a general means of adding cost to the objective without it becoming the overriding consideration in the problem as our first concern should be obtaining the capabilities needed by the Canadian Forces. Perhaps the

best way to deal with the situation above would be to have a goal level of 3 for the field hospital in period 3 instead of 2. If this was combined with an appropriate threshold level and a small penalty weight for not meeting the goal, the model could probably be coerced into maintaining level 3 in period 3, but with a high degree of flexibility should the funds be needed elsewhere (in much the same way that in the sample problem the model chose levels that gave seemingly unnecessary deviations in time periods 1 and 3 because it was more important to have the necessary funds in time period 4). This approach to the problem highlights a point that has yet to be addressed. Should a computer means be used to generate and automatically feed goal levels into the algorithm, a second run would be required with a manual override of the levels. This is consistent with the author's view that no single run of the model will produce "the" final result. Rather, it is envisioned that multiple runs will be required, with an analysis of each run suggesting modifications to the goal levels and/or weight terms for subsequent runs. The use of multiple runs will not only assist in showing the robustness of a given result, but should also appease any concerns the various environments have with the model's output, particularly in the case of goal sets that are initially generated by computer means.

37. A second possible means of dealing with the fluctuating goal levels would be to use the preemptive Goal Programming approach alluded to above. One could use the present algorithm to identify those capability mixes that not only best meet the desired goals, but also those that exceed the goal and are still within the budgetary constraints. The results could then be fed into a second Linear Programming program that tries to minimize costs. Such a scheme would also be useful when the dynamic costing makes it worthwhile to consider obtaining excess capability levels today with a view to tomorrow.

38. As a final note, Dynamic Programming may very well be used in the end as the means of implementing the Goal Programming methodology instead of the Linear Programming formulation presented here. Linear Programming is limited by the fact that the objective function and constraints imposed must, of course, be linear equations. As the Force Planning Scenarios project develops, concepts that lend themselves better to nonlinear modeling may arise. The Dynamic Programming methodology outlined in Ref. 4 has the advantage that it is based in low-level computer programming, at least in

comparison to the use of a pre-packaged Linear Programming routine, and easily accomodates nonlinear functions in its implementation.

Conclusions

39. In this research note we have attempted to outline Goal Programming as a candidate methodology for use in the long-term strategic planning project currently underway. While the methodology will not provide answers to all of the issues faced in the project, it can provide insight into particular aspects of the study. The Goal Programming algorithm put forward is still very much in the development stage, with amendments forthcoming as the long-term strategic planning project itself advances toward completion.

References

1. Bradfield, A., Christopher, G., MacLean, LCol D., The Development of a Scenario Set for Departmental Force Planning, DOR(J&L) Research Note RN 9822, November, 1988.
2. Hillier, F.S., Lieberman G.J., Introduction to Operations Research, Fourth Edition, Holden-Day, Inc., Oakland CA, 1986.
3. Ragsdale, C.T., Spreadsheet Modeling and Decision Analysis, South-Western College Publishing, Cincinnati, OH, 1998
4. Isbrandt, S., A Dynamic Programming Concept For Strategic Planning, DOR(J&L) Research Note RN 9819, November, 1998.
5. Fourer, R., Gay, D.M., Kernighan, B.W., AMPL A Modeling Language For Mathematical Programming, Boyd & Fraser Publishing Company, Danvers, MA, 1993.

Goal Programming Methodology

Found herein is an AMPL Plus® (Ref. 5) implementation of the Goal Programming algorithm developed in this research note. This ANNEX details the model itself including the objective function and constraints, as well as the declaration of all variables and paramters used. ANNEX B outlines the paramter values used in the given example. Note that the use of a #-symbol is indicative of a comment in the code.

```
#
# Force Planning Scenarios Goal Programming Algorithm
# Date: November, 1998
#

param TotalCaps;                          # Total number of capabilities
param TotalPeriods;                       # Total number of periods under consideration

set CAP:=1..TotalCaps;                    # Indexing set for the capabilties
set PERIOD:=1..TotalPeriods;              # Indexing set for the periods

param StartLevel{CAP};                    # Starting Level for a given capability
param MaxLevel{CAP};                      # Maximum capability level

set LEVEL{c in CAP}:=0..MaxLevel[c];      # Indexing set for the capability levels
```

```
param ActualValue{c in CAP,LEVEL[c]};                      # Actual value of a capability level

param GS{CAP,PERIOD} integer;                              # Goal Set

param ALPHA{CAP} > 0;                                      # Utility function parameter
param BETA{CAP} > 0;                                       # Utility function parameter

param GS{CAP,PERIOD} integer;                              # Goal Set

param Utility{c in CAP,i in LEVEL[c]}:=1-exp(-ALPHA[c]*((ActualValue[c,i])^BETA[c]));
param GSUtility{c in CAP,p in PERIOD}:=1-exp(-ALPHA[c]*((ActualValue[c,GS[c,p]])^BETA[c]));

param T{c in CAP,p in PERIOD} default GS[c,p];             # Threshold for negative deviations
param Cmin{c in CAP,p in PERIOD}:=GS[c,p]-T[c,p];          # Minimum Purchase Limit
param Cmax{c in CAP} default max{p in PERIOD} GS[c,p];     # Maximum Purchase Limit (derived from Goal Set)


param pwt{PERIOD};                                         # Weight term for periods
param wm{CAP,PERIOD};                                      # Penalty Weight for negative deviations
param wp{CAP,PERIOD} default 0.001;                        # Weight for excess deviations (set to a token small +ve value)
param BB{PERIOD};                                          # Base Budget for a given period
param INT{2..TotalPeriods};                                # Interest accrued on unspent monies

var WD{PERIOD};                                            # Weighted deviation measure
var dm{CAP,PERIOD};                                        # Negative deviation from Goal Set
var dp{CAP,PERIOD};                                        # Excess deviation from Goal Set
var B{PERIOD};                                             # Budget in a given period
var C{PERIOD};                                             # Cost in a given period
var I{c in CAP,LEVEL[c],LEVEL[c],PERIOD} binary default 0; # Indicator variable between nodes in the network
```

```
        # Objective function

minimize Sum_Cap_Deviations:  sum{p in PERIOD} pwt[p]*WD[p];
                #Sum of period capability deviations weighted by period


        # Constraints

subject to Weighted_Deviations {p in PERIOD}: WD[p]=sum{c in CAP:GSUtility[c,p]>0}
(wm[c,p]*dm[c,p]+wp[c,p]*dp[c,p])/GSUtility[c,p];
        # Weighted deviation measure

subject to Demand {c in CAP, p in PERIOD}: sum{i in LEVEL[c],j in LEVEL[c]} I[c,i,j,p]*Utility[c,j]
        +dm[c,p]-dp[c,p]=GSUtility[c,p];
        # Function to determine the deviations dm and dp

subject to Min_dm {c in CAP, p in PERIOD}: dm[c,p]>=0;
        # Non-negativity constrant for negative deviations

subject to Min_dp {c in CAP, p in PERIOD}: dp[c,p]>=0;
        # Non-negativity constraint for excess deviations

subject to Min_Obtain {c in CAP, p in PERIOD}: sum{i in LEVEL[c],j in LEVEL[c]} I[c,i,j,p]*j >= Cmin[c,p];
        # Must purchase at least the minimum capability level required
```

```
subject to Max_Obtain {c in CAP, p in PERIOD}: sum{i in LEVEL[c],j in LEVEL[c]} I[c,i,j,p]*j <= Cmax[c];
        # Must not purchase more than the maximum capabiltiy level allowed

subject to Start {c in CAP, i in LEVEL[c]:i=StartLevel[c]}:  sum{j in LEVEL[c]} I[c,i,j,1]=1;
        # Starting from the base level, only one of the links to a node in period 1 can be reached.
        # i.e., The sum of binary variables equalling 1 means that all but one equal zero.

subject to Start2 {c in CAP,i in LEVEL[c]:i<>StartLevel[c]}: sum{j in LEVEL[c]} I[c,i,j,1]=0;
        # Constraint to ensure the path begins at the base level

subject to Balance {p in 2..TotalPeriods,c in CAP,i in LEVEL[c]}: sum{j in LEVEL[c]} I[c,i,j,p]=
        sum{k in LEVEL[c]} I[c,k,i,(p-1)];
        # Balancing constraint to ensure that where we go from period (p-1) to period p
        # begins from where we are in period (p-1)

subject to Cost_Constraint {p in PERIOD}: C[p] <= B[p];
        # Cannot spend more in a period than the budget allows

subject to Start_Budget: B[1]=BB[1];
        # Start budget

subject to Subsequent_Budget {p in 2..TotalPeriods}: B[p]=BB[p] + (B[(p-1)] - C[(p-1)])*INT[p];
        # Subsequent budgets are determined from the period's base budget plus any monies
        # left over from previous periods

subject to Spent {p in PERIOD}: C[p]=sum{c in CAP,i in LEVEL[c],j in LEVEL[c]} CC[c,i,j,p]*I[c,i,j,p];
        # Sum of flagged capability costs for a given period
```

ANNEX B
DOR(J&L) RN 9824
DECEMBER 1998

Sample Problem Data

Found herein is the parameter data used in the sample problem, formatted as required for use by AMPL Plus®. The software facilitates the use of matrices for such data. For a parameter with two indices, the rows of the matrix correspond to the first index while the columns equate to the second. For parameters with more indices, such as the capability costing parameters, all but two must be specified for a given matrix, the rows and columns corresponding to the first and second unspecified indices respectively.

```
param TotalCaps:=3;                         # Total number of Capabilities
param TotalPeriods:=4;                      # Total number of Time Periods

param GS:                                   # Goals for each Capability-Period
        1     2     3     4:=               # combination
1       3     3     3     3                 # Each column relates to a Period
2       1     1     2     3                 # Each row relates to a Capability
3       1     1     1     1;

param T :                                   # Minimum level thresholds for each
        1     2     3     4:=               # Capability-Period combination
1       1     1     1     1                 # Note the default value provided in the
2       0     1     1     0                 # model.
3       1     1     0     1;                # Each column relates to a Period
                                            # Each row relates to a Capability

param pwt:=                                 # Period weights
1       1                                   # Each row relates to a Period
2       1
3       2
4       4;

param MaxLevel:=                            # Maximum levels for the Capabilities
1       3                                   # Each row relates to a Capability
2       3
3       3;
```

```
param StartLevel:=                          # Period 0 levels for the Capabilities
1     1                                     # Each row relates to a Capability
2     1
3     1;

param ALPHA:=                               # Utility function parameter
1     1                                     # Each row relates to a Capability
2     1
3     1;

param BETA:=                                # Utility function parameter
1     1                                     # Each row relates to a Capability
2     1
3     1;

param ActualValue:                          # Quantitative values for each Capability-
      0     1     2     3:=                 # Level combination
1     0     1     2     3                   # Each column relates to a Capability level
2     0     1     2     3                   # Each row relates to a Capability
3     0     1     2     3;

param INT:=                                 # Interest on unspent funds
2     1.1                                   # Each row relates to a Period
3     1.1
4     1.1;

param BB:=                                  # Base Budget for each Time Period
1     10                                    # Each row relates to a Period
2     10
3     10
4     10;

param wm:                                   # Penalty Weights for each Period-
      1     2     3     4:=                 # Capability combination
1     1     1     1     1                   # (Underachievement of goals).
2     1     1     1     1                   # Each column relates to a Period
3     1     1     1     1;                  # Each row relates to a Capability

# Weights are not specified in the data set for overachieving the goals.  These are given
# default values in the model itself.
```

```
param CC :=                                  # Capability Cost for capability [i,*,*,*]
[1,*,*,1]:   0    1    2    3:=              # to make the transition from level [*,j,*,*]
       0     0    3    5    8                # to level [*,*,k,*] in period [*,*,*,l]
       1     0    0    4    9                # Each column relates to the acquired
       2     0    0    1    5                # Capability level
       3     0    0    2    2                # Each row relates to the existing
                                             # Capability level

[2,*,*,1]:   0    1    2    3:=
       0     0    7    12   18
       1     0    0    10   18
       2     0    0    5    20
       3     0    0    0    10

[3,*,*,1]:   0    1    2    3:=
       0     0    5    10   15
       1     0    0    5    20
       2     0    0    2    8
       3     0    0    0    5

[1,*,*,2]:   0    1    2    3:=
       0     0    3    5    8
       1     0    0    4    9
       2     0    0    1    5
       3     0    0    2    2

[2,*,*,2]:   0    1    2    3:=
       0     0    7    12   18
       1     0    0    10   18
       2     0    0    5    20
       3     0    0    0    10

[3,*,*,2]:   0    1    2    3:=
       0     0    5    10   15
       1     0    0    5    20
       2     0    0    2    8
       3     0    0    0    5

[1,*,*,3]:   0    1    2    3:=
       0     0    3    5    8
       1     0    0    4    9
       2     0    0    1    5
       3     0    0    2    2
```

[2,*,*,3]:	0	1	2	3:=
0	0	7	12	18
1	0	0	10	18
2	0	0	5	20
3	0	0	0	10

[3,*,*,3]:	0	1	2	3:=
0	0	5	10	15
1	0	0	5	20
2	0	0	2	8
3	0	0	0	5

[1,*,*,4]:	0	1	2	3:=
0	0	3	5	8
1	0	0	4	9
2	0	0	1	5
3	0	0	2	2

[2,*,*,4]:	0	1	2	3:=
0	0	7	12	18
1	0	0	10	18
2	0	0	5	20
3	0	0	0	10

[3,*,*,4]:	0	1	2	3:=
0	0	5	10	15
1	0	0	5	20
2	0	0	2	8
3	0	0	0	5 ;

DOCUMENT CONTROL DATA (Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)	
1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared e.g. Establishment Sponsoring a contractor's report, or tasking agency, are entered in Section 8). Operational Research Division Department of National Defence Ottawa, Ontario K1A 0K2	2. SECURITY CLASSIFICATION (overall security classification of the document, including special warning terms if applicable) UNCLASSIFIED
3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title) **GOAL PROGRAMMING FOR STRATEGIC PLANNING (U)**	
4. AUTHORS (last name, first name, middle initial) Mr. R. Dickson	
5. DATE OF PUBLICATION (month Year of Publication of document) DECEMBER 1998	6a. NO OF PAGES (total containing information. Include Annexes, Appendices, etc.) 35 6b. NO OF REFS (total cited in document) 5
7. DESCRIPTIVE NOTES (the category of document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.) Directorate of Operational Research (Joint & Land) Research Note	
8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address). VCDS/DGSP/DDA	
9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)	9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written.)
10a. ORIGINATOR's document number (the official document number by which the document is identified by the originating activity. This number must be unique to this document.) DOR(J&L) Research Note RN 9824	10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor.)
11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification.) (X) Unlimited distribution ( ) Distribution limited to defence departments and defence contractors: further distribution only as approved ( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved ( ) Distribution limited to government departments and agencies; further distribution only as approved ( ) Distribution limited to defence departments; further distribution only as approved ( ) Other (please specify):	
12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)	

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the test is bilingual).

This research note outlines how a Goal Programming methodology could be used as an analysis tool for scenario-based strategic planning of the Canadian Forces. A sample problem is put forward and solved in a Linear Programming framework. Implementation details and computer code for the algorithm are provided in Annex sections of this note.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus, e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified . If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

ANALYSIS
GOAL PROGRAMMING
OPTIMIZATION
STRATEGIC PLANNING
CAPABILITY PLANNING
FORCE DEVELOPMENT

Canada

#509885