A Hybrid Tabu Search/Branch and Bound Approach to Solving the Generalised Assignment Problem.

Introduction

A number of effective solution methods for the classical Generalised Assignment Problem (GAP) have been proposed during the last thirty years. Exact branch and bound approaches to solving GAP have been proposed by (Ross, Soland 1975, Martello, Toth 1990). More recently however heuristics have been utilised in order to reduce the size of the branch and bound tree by seeking good feasible solutions such as (Nauss 2003). In contrast to the branch and bound approach some highly efficient heuristic solution methods for GAP have been developed during the last ten years including (Beasley, Chu 1997, Diaz, Fernandez 2001, Laguna, Kelly et al. 1995, Narciso, Lorena 1999, Osman 1995, Yagiura, Ibaraki et al. 2004).

GAP is one of the representative combinatorial optimisation problems known to be NP-hard and can be stated as the problem of optimising the assignment of *n* jobs to *m* agents, where *n* is usually much larger than *m*, such that each job is assigned to exactly one agent and the resource capacity of each agent is not violated. Standard libraries of test problems exist that have been used in order to compare the effectiveness of alternative solution methods. The difficulty of these problem instances has been categorised into five classes A, B, C, D and E with classes D and E being generated as a result of the more recently developed methods requiring harder problems in order to demonstrate their effectiveness. Classes A, B and C become progressively more difficult due to tighter constraints on the problem and classes D and E provide an added aspect of difficulty since the cost of assigning jobs to agents is inversely related to the resource required for such an assignment.

The motivation for this study is the success of Tabu Search (TS) approaches to solving GAP along with ideas due to (Glover, Laguna 1997) such as *Tabu Branching Moves* and *Referent Domain Optimisation*, and *Dynamic Branch and Bound* (Hanafi, Saïd - Glover, Fred 2002). The approach taken in the research is one of utilising existing commercially available software (Xpress-MP) in order to provide an LP based Branch and Bound aspect and combining this with a TS strategy that interacts with the branch and bound aspect in order to produce a hybrid solution method for GAP.

Tabu Branching Moves

Tabu Branching Moves (TBM) are proposed in (Glover 1990) and(Glover, Laguna 1997). There are three types of move proposed by Glover and Laguna which will be described below that can be used to construct a path from the root node of a branch and bound tree to one of the leaf nodes of the tree. The solution at each node of the tree is produced by solving the LP relaxation of the problem and the choice of branching variable can be decided using techniques normally associated with traditional branch and bound methods. The three types of TBM are as follows:-

Restriction Moves

This type of move is exactly the same as that which would be employed in a traditional branch and bound scheme and involves imposing a new branch on the tree by branching either up or down on one of the fractional variables in the solution at the current node.

Reversal Moves

This type of move involves dropping a previously imposed branch from the current sequence of branches that define the path through the tree and then immediately imposing its complement. This type of move can only be achieved within a branch and bound scheme where a branch meets one of the end nodes of the tree and is then reversed to branch on its complement. A degree of flexibility can be achieved from a tabu branching scheme that allows the reversal of branches that were previously imposed as part of the current sequence. This is achieved by dropping the branch from the sequence and subsequently adding its complement branch to the end of the collapsed sequence thus forming a new sequence and generating the new end node by solving the LP relaxation according to the new sequence of restrictions. As a result of this TBM are able to maintain the depth of the search at a level that is likely to find integer feasible end nodes, infeasible end nodes or nodes whose LP solution values exceed the best integer solution found thus eliminating the need to extend the current sequence.

Relaxation Moves

This type of move effectively collapses the current sequence of branches by removing a branch that was imposed earlier in the search by means of a restriction move. This is in contrast to traditional branch and bound which can only delete previously imposed restrictions in a sequential manner as part of a backtracking or jump tracking strategy.

The implementation of a TBM scheme using the three types of moves described above requires the construction of a set of rules to determine move selection at each step along with a tabu management structure in order to avoid the regeneration of previous sequences and prevent the method from cycling. The method currently constructed as part of this research first uses a restriction phase that follows a depth first approach until reaching a node that would result in a backtracking step in a normal branch and bound search. The trajectory of the search is then changed either by a reversal move, or if no non tabu reversal move is available then a relaxation move is made. As a result the branch previously dropped from the sequence either as part of a reversal or as a relaxation move becomes tabu active thus preventing it from being reimposed as a restriction and the restriction phase is then reimplemented to produce a new sequence. As suggested in (Glover, Laguna 1997) TBM can be used either as part of a Branch and Bound scheme or as a supplement to it by generating trial solutions as part of the branch and bound strategy.

The flexibility allowed by the use of TBM allows for the maintenance of solution sequences that include highly influential branches whilst at the same time allowing the process to drop uninfluential branches previously imposed. This is in contrast to branch and bound that must live with previously imposed decisions however poor they subsequently turn out to be. The issue of quality and influence within a branch and bound scheme is discussed in (Hanafi, Saïd - Glover, Fred 2002) where it is shown that by taking advantage of information gained from branches at deeper levels of the tree a branch and bound search for an optimal solution may be obtained sooner by changing the structure of the tree based on information obtained by branching on certain variables. This dynamic construction and links with TBM is now described.

Dynamic Branch and Bound

Dynamic Branch and Bound (DBB) was first proposed in (Glover, Tangedahl 1976) and has been more recently discussed in (Hanafi, Saïd - Glover, Fred 2002) where it is suggested that DBB may have potential that as yet remains largely unexplored. The dynamic aspect of DBB is provided by two processes, these being

i. Shrinking.
ii. Resequencing.

A worked example demonstrating how these two processes can affect the branch and bound process is given in (Hanafi, Saïd - Glover, Fred 2002) but is essentially concerned with using information gained from later branching decisions in order to reconstitute the tree to produce another tree from which an optimal solution may be found sooner than would otherwise have occurred. The process employs standard branch and bound techniques to construct a tree to the point where backtracking would normally occur at which point the opportunity to shrink is taken by eliminating branches from the tree that have been subsequently rendered useless by the imposition of a branch imposed later thus pruning the tree whilst maintaining the more influential branches. Further to shrinking the tree it is then shown that by resequencing the order of the branches in the remaining tree it is possible to maintain the most influential branches near to the root of the tree. A definition of what constitutes an influential branch as opposed to a non-influential branch is given in the same paper. By maintaining a sequence of influential branches which can be constructed using the types of branching moves described previously it is then possible to generate an end node whose LP relaxation is highly restrictive and as such is more likely to be a node that requires no further branching.

By maintaining a sequence of branches in order of influence, the highest influence branches being those closer to the beginning of the sequence, and shrinking the sequence when it cannot be further extended will define the region that can be searched from the end of this sequence.

By generating sequences that restrict a problem in terms of fixing variables using a heuristic method it is then possible to optimise the restricted problem in order to find a feasible integer solution or to rule out the sequence by determining that no feasible solution exists by optimising the restricted problem. This is described in (Glover, Laguna 1997) as *Referent Domain Optimisation* and is also referred to in (Budenbender, Grunert et al. 2000).

Referent Domain Optimisation

Referent domain optimisation is referred to by Glover and Laguna as being a process of using optimisation methods together with heuristic processes to generate trial solutions which can then be optimised to generate a new solution and the process repeated. Several examples of how this can be achieved are given. In the context of this research TBM are used to generate trial sequences as described previously and a branch and bound procedure can then be used to optimise the restricted problem defined by the sequence generated by the heuristic.

The strategy being developed is in keeping with an observation made in (Glover, Laguna 1997) which is based on the fact that a problem with a small number of zero-one variables can typically be solved quite quickly using a branch and bound solver such as Xpress-MP. The idea is to fix a large number of the problem variables using TBM, if after generating a trial sequence the resulting restricted LP problem has less than *r* free variables, where *r* is a parameter of the method, then the problem can be post-optimised using branch and bound. As a result of optimising the restricted problem a feasible integer solution may or may not have been found and it is then necessary to continue the search by generating a new trial sequence to be optimised. Tabu memory structures are required in order to prevent cycling and to provide diversification strategies as described subsequently.

Algorithm

Step 1.
Initialise the solver.

Step 2.- Restriction Phase.

i. Solve the LP relaxation.
ii. Select branching variable.
iii. Branch up on selected variable and go to i. If the selected variable cannot be fixed to 1 (according to the resource requirement for that assignment and the available resource for the agent) then branch down on the selected variable and go to i. If the selected branch cannot be fixed to 0 (the resulting LP relaxation is infeasible) then go to step 3.

Step 3.
Attempt to shrink the current sequence.

Step 4.
Resequence.

Step 5.

Reverse a branch in the current sequence and go to step 7. If no feasible reversals exist then go to step 6.

Step 6.
Relax branches in the current sequence until the resulting solution is LP feasible and then go to step 7.

Step 7.
If the number of fixed variables is greater than *r* then optimise using branch and bound, record the best solution found and go to step 5, else go to step 2.

The algorithm described above can be run for a fixed number of iterations or until a specified number of iterations have elapsed without the best known solution improving, at which point it is then necessary to diversify the search in some way in order to generate sequences that are significantly different to those generated previously.

Tabu Restrictions

The tabu status of a branch that is added to the sequence must prevent that branch from either being reversed or dropped by a relaxation move until t reversal or restrictions have been made since that branch was added. Since a reversal move effectively consists of dropping a branch from the sequence and then adding its complement to the sequence then the dropping of a branch that has been added to the sequence is dependant upon the number of branches subsequently added for its tabu status. A branch that has been dropped either as a relaxation move or as part of a reversal move must then be prevented from being included in the sequence and as such is also dependant on the number of branches subsequently added for its tabu status. These types of restrictions require the use of a counter k_{add} that is only increased each time a restriction or reversal move is applied. This is in contrast to the suggestion in (Glover 1990) of using three separate tabu lists to apply to each type of move.

Diversification of the search may be achieved by performing one or a series of relaxation or reversal moves. Depending upon the level of diversification required at different stages it may be necessary to remove highly influential and restrictive branches and prevent them from being included in the current sequence for a number of iterations. It is thought that a critical event memory of the type described in (Glover, Laguna 1997) that uses measures of frequency in order to introduce solution attributes not previously included and exclude attributes that have high residence frequencies, would be one way of achieving such diversification. Alternative frequency based memory structures may also be considered.

Summary

A hybrid Tabu Search/Branch and Bound method for solving GAP will be developed by using TBM in order to generate sequences with end nodes that represent restricted LP relaxations of the original problem. If a sequence contains a pre-determined number of fixed variables then the branch and bound solver will be called in order to optimise the restricted problem. The concept of shrinking and resequencing proposed by Glover in relation to Dynamic Branch and Bound will also be used in order to restructure sequences which will have the effect of reducing the size of the reverse and relax neighbourhoods. A new trial sequence is then generated and the process repeats. A series of non-improving iterations will instigate a diversification phase that will drive the search into new, unexplored regions. Both short and long term memory structures will be used in order to guide the overall strategy. The results of some preliminary computational work will also be presented. It is ultimately envisaged that the whole process can be embedded within the Xpress-MP software in order to take full advantage of the branch and bound solver.

References

BEASLEY, J.E. and CHU, P.C., 1997. A Genetic Algorithm for the generalised assignment problem. *Computers and Operations Research,* **27**, pp. 17-23.

BUDENBENDER, K., GRUNERT, T. and SEBASTIAN, H.-., 2000. A Hybrid Tabu Search/Branch-and-Bound Algorithm for the Direct Flight Network Design Problem. *TRANSPORTATION SCIENCE,* **34**(4), pp. 364-380.

DIAZ, J.A. and FERNANDEZ, E., 2001. A Tabu search heuristic for the generalized assignment problem. *European Journal of Operational Research,* **132**, pp. 22-38.

GLOVER, F. and TANGEDAHL, L., 1976. Dynamic Strategies for Branch and Bound. *Omega,* **4**, pp. 571-576.

GLOVER, F., 1990. Tabu Search-PartII. *ORSA Journal on Computing,* **2**(1), pp. 4-32.

GLOVER, F. and LAGUNA, M., 1997. Tabu search. Boston: Kluwer Academic Publishers.

HANAFI, SAÏD - GLOVER, FRED, 2002. Resolution Search and Dynamic Branch-and-Bound. *Journal of Combinatorial Optimization,* **6**(4), pp. 401-424.

LAGUNA, M., KELLY, J.P., GONZALEZ-VERLADE, J.L. and GLOVER, F., 1995. Tabu search for the multilevel generalized assignment problem. *European Journal of Operational Research,* **82**, pp. 176-189.

MARTELLO, S. and TOTH, P., 1990. Knapsack Problems: Algorithms and Computer Implementations. Chichester: Wiley.

NARCISO, G.M. and LORENA, L.A.N., 1999. Lagrangean/surrogate relaxation for the generalized assignment problem. *European Journal of Operational Research,* **114**, pp. 165-177.

NAUSS, R.M., 2003. Solving the Generalised Assignment Problem: An Optimizing and Heuristic Approach. *INFORMS Journal on Computing,* **15**(3), pp. 249-266.

OSMAN, I.H., 1995. Heuristics for the Generalised assignment problem: Simulated Annealing and Tabu Search approaches. *OR Spektrum,* **17**, pp. 211-215.

ROSS, G.T. and SOLAND, P., 1975. A Branch and Bound based algorithm for the generalised assignment problem. *Mathematical programming,* **8**, pp. 91-103.

YAGIURA, M., IBARAKI, T. and GLOVER, F., 2004. An ejection chain approach for the generalized assignment problem. *INFORMS Journal on Computing,* **16**(2), pp. 131-151.