EUCLID CALMA
Radio Link Frequency Assignment Project
Report 2.2.1: Implementation and Testing of Polyhedral Techniques and Interior Point Methods

K.I. Aardal
A. Hipolito
C.P.M. van Hoesel
B. Jansen
C. Roos
T. Terlaky

T.U. Eindhoven RLFAP Group
T.U. Delft RLFAP Group

1 Object

Many military decisions can be formulated as combinatorial optimization problems. Such examples include

- weapons-targets assignments,
- areas to military units assignments,
- cargo-loading,
- convoy routing,
- radio link frequency assignment,
- and many others.

Furthermore, many of these decisions are still made by humans, often without resource to computing power and usually adhering to some crude algorithm which has evolved over many years.

The past decade has witnessed both the development of many promising optimisation algorithms as well as the continuing development of computing technology to the extent that it is now feasible to solve problems of the magnitude of some real world instances, within reasonable processing time constraints.

Combinatorial optimization problems can be tackled by several approaches. So far, there is general agreement on the fact that it has not yet been proved that any one of these approaches is more efficient than any other for the general case. Thus, for each problem type, the choice of which approach to use to best solve the problem remains open.

A comparative study of the various solution approaches for every problem type is therefore required, to give guidance to practitioners. However, scientific characterization of all combinatorial problems in order to be able to specify which approach is the most relevant for solving all the problems of a given class, would need significant theoretical advances. Moreover the choice of an approach is frequently based on several criteria (accuracy, speed, robustness, development cost, etc) and it is well known that multi-criteria based decisions are vey difficult. Nevertheless, some empirical rules based mainly on experimental results, but supported in part by theoretical analyses, should provide guidance for the decision makers in their choice of approach for any given problem type.

Moreover, one of the most significant outcomes of the CALMA project will be guidance for the efficient development of each of the chosen approaches.

2 Concern

2.1 Global Objectives

In this scientific and operational context, the following steps have been proposed as part of the EUCLID program (CEPA6-RTP 6.4).

1. To solve a set of problems of the same type by various approaches.
2. To determine, according to the obtained results, the weakness, strengths and potential of each approach.
3. To compare the obtained results according to a well accepted comparison method.
4. To synthesize these results in a method for selecting the most promising approach for new problem instances.
5. To study in depth several other military applications of combinatorial algorithms and to propose the most efficient approach according to the rules exhibited in the previous steps.
6. To disseminate the acquired knowledge among the Defence community of the participating countries.

A case study has been selected by the management group of the RTP 6.4: it is the *Radio Links Frequency Assignment Problem*, or *RLFAP*.

2.2 Precise objective: *Work Element 2.2.1–Implementing and testing polyhedral techniques and interior point methods*

This work element aims at using recent advances in algorithm development for integer linear programming formulations of hard combinatorial optimization problems. The objective is to determine optimal solutions, or, if that is not possible, generate good lower bounds along with good feasible solutions, in order to obtain a good estimate of the value of optimal solutions within reasonable time.

Our approach is based on solving the linear programming relaxation of an initially integer linear programming formulation (ILP) of the RLFAP. In essence, the relaxation is constructed by removing integrality constraints from the formulation. Thus, the set of feasible solutions to the problem is extended with fractional solutions satisfying the linear constraints. Since the problems have an objective which is to be minimized, the value of the optimal solution of the linear programming relaxation is a lower bound on value of the original problem. Such a lower bound, together with a feasible solution of the original problem provide an interval in which the optimal value is guaranteed to be found, since the value of any feasible solution is an upper bound on the optimum value of a problem.

A major part of our efforts in solving the problem is put in strengthening the linear programming relaxation of the RLFAP, by searching for additional linear constraints that can be added to the linear programming relaxation. This process is called a **cutting plane** algorithm. It is embedded into a tree search algorithm together with various other techniques, which is known as the **branch and cut** framework. The tree search algorithm partitions a problem into subproblems recursively, by fixing variables at one of their possible values. It involves not only the selection of a variable, but also making the choice of which value to examine first. These selection decisions constitute the branching rule in the branch and cut process.

The components of the framework include, besides the components mentioned above, various techniques. The techniques that are incorporated into our final algorithm are described below.

- Lower bounding. Linear constraints are obtained from specific structures in the instances of the problem.

- Upper bounding. Heuristic methods based on the linear programming relaxation are capable of finding good solutions. In combination with the lower bounds the search process is substantially reduced.

- Efficient solving of linear programs. The branch and cut framework may involve solving many linear programs of large size that have to be solved efficiently using advanced simplex or interior point methods.

- Branching rules for creating subproblems and selection of subproblems form the basis of the search process in the framework. The search tree in a branch and cut framework contains a large list of subproblems. Maintenance of the linear constraints is a complicated matter in the process.

- Preprocessing to improve and reduce problem formulations. This is highly effective for the initial problem, and is often useful in intermediate stages. Basically, we try to remove links from an instance if they can be assigned a frequency that does not increase the value of the solution.

This work element seeks to implement these components to solve the radio link frequency assignment problem with branch and cut.

2.3 Brief history

The branch and cut algorithmic framework is considered to be one of the state-of-the-art approaches for solving hard combinatorial optimization problems such as the traveling salesman problem (TSP), the linear ordering problem and the node-packing problem (which is very close to the frequency assignment problem). A recent survey of the approach and a list of successful practical applications using it can be found in Aardal and van Hoesel [1]. An overview of implementational issues of branch and cut can be found in Jünger et al. [4]. A generic branch and cut code (MINTO) for integer linear programming problems has been developed by Nemhauser et al. [9]. One of the main features of this package is that it provides the possibility of incorporating problem-specific routines (programmed in C) and integrate them into the standard branch and cut framework.

Problems related to the frequency assignment problem are the node packing and node coloring problems on graphs. The branch and cut approach to these problems has obtained very little attention, so far. We can only mention a study by Nemhauser and Sigismondi [10], and a study of the colouring problem using column generation, see the DIMACS challenge.

Linear programming plays an important role in our branch and cut framework. After the development of Karmarkar's interior point algorithm in 1984 [6], we have seen an upsurge of developments in solving linear programs. This has also led to a push in greater efficiency in variants of the simplex method. One of its consequences is that increasing sizes of integer programs can now be tackled. Recently, investigations in using interior point methods for integer programming have indicated some promise (e.g. Borchers and Mitchell [3]).

2.4 Acquired results

One successful application of the branch and cut framework has been the traveling salesman problem TSP. The largest instance of the TSP that has been provably solved to optimality has 7397 cities (Applegate, Bixby, Chvatal and Cook). In addition Jünger, et al. [5] have also demonstrated how to find reasonably good solutions as well as lower bounds for large instances (from 100 to 4461 cities) of the TSP within given computational time limits using the branch and cut framework.

Our interest in application of the branch and cut framework comes from the node packing problem. This has been investigated in Nemhauser and Sigismondi [10], who report to have solved node packing problems up to 120 variables. The study of the coloring problem using column generation, also reports on solving only small instances of up to 200 variables.

Regarding the solution of linear programs, the state of the art has been documented in Bixby [2] for simplex methods and in Lustig, et al. [7] for interior point methods. These results show that there are specific types of linear programs for which one method may be more suitable than the other. This work element will show in part which method may be suitable for the LP relaxations associated with the RLFAP.

3 Study progress report

3.1 Specific information

Problem preprocessing and initial lower bounding were conducted by the TU Eindhoven group and these results were used by the TU Delft group in an implementation of a specialized branch and cut code for the RLFAP. Discussions on strategies to use with respect to valid inequalities, and branching rules were conducted between the two groups.

The specialized branch and cut code was developed using the MINTO 2.0 package. This package provides a shell with which problem-specific routines can be programmed in C and integrated into a branch and cut algorithm. The CPLEX 3.0 LP package was used for the LP-solving (simplex and interior point) within MINTO.

3.2 Reminder of the work breakdown

The initial proposal breaks the work down into the following topics: branch and bound, branch and cut algorithms, cutting plane methods, polyhedral techniques, interior point methods. The second topic is a specialization of the first, with regard to the used relaxation, namely linear, of the problem. The third and fourth topic are related to strengthening linear programming relaxations with valid linear inequalities. The interior point methods may be used as solvers for the linear programs.

During the course of working on this work element, we have found that it is best to structure the work in a single branch and cut framework which incorporates the following components. Each of the components can be put in the framework separately. Nevertheless, several persons cooperated on more than one component.

1. Preprocessing to reduce the size of the initial formulation.
2. Lower bounding by generation of valid inequalities.
3. Efficient constraint management (generation and deletion).
4. Upper bounding by heuristic generation of primal feasible solutions.
5. Branching and node selection.
6. Efficient LP-solving using advanced simplex and/or interior point methods.

3.3 Obtained results and significant points

3.3.1 Results on preprocessing

The RLFAP problem consists of a set of links that have to be assigned frequencies. With different dominance criteria one can prove that under the objective of minimizing the number of used frequencies some links can always be assigned a feasible frequency. These links are therefore removed from the initial problem instance. The percentage reduction in the number of links is given in Table 1 below, in the third column. This reduction can be used not only for our techniques, but for any technique for the RLFAP. The instances have been reduced even more by removing links for which is it likely that they can be assigned a frequency. The total percentage reduction can be found in the fifth column of the table below. We used the reduced instances to solve the problem instances 1, 2, 3 and 11. In the last column of the table one can find the CPU times (on a 486-66) for performing the reductions in seconds.

3.3.2 Results on lower bounding techniques 1

The clique number of the instances, where all frequency distances have been reduced to 1, gives a lower bound on the number of frequencies that have to be used. With a simple branching rule which states that two links are assigned either the same frequencies or different frequencies this lower bound can be increased. The results are found in the Table 2. The

Instance	Size	Reduced size 1	Time 1	Reduced size 2	Time 2
1	916	806	5 sec	298	8 sec
2	200	166	1 sec	78	1 sec
3	400	348	2 sec	212	3 sec
11	680	646	3 sec	194	11 sec

Table 1: Preprocessing results.

bounds below were obtain in less than ten minutes computation time for 1 and 3, while a few seconds for 2 and 11 sufficed.

Instance	Clique bound	Improved bound
1	12	14
2	14	14
3	12	14
11	20	20

Table 2: Lower bounding results I.

3.3.3 Results on lower bounding techniques 2

We have also obtained lower bounds for the second type of frequency assignment problems, namely those in which we want to minimize the weighted sum of interferences. The idea is again simply to solve another combinatorial problem, the optimal value of which is lower than the original one, and the optimal solution of which can be computed efficiently.

The instances with pre-assigned frequencies can be relaxed in the following sense: instead of selecting for each link the 'best possible' frequency, we only decide whether or not to stick to the preassigned frequency. In case we use the preassigned frequency we can use this knowledge to compute almost exactly the interference cost in this solution. On the other hand, if we decide not to use the preassigned frequency we only know for sure that mobility cost is incurred, but we can only estimate the amount of interference associated with our alternative choice.

The problem that arises is solved effectively by enumerating all possible decisions. We designed an efficient way of going through all relevant possibilities and in the end, we were able to solve these relaxations in a matter of seconds. This allowed for the use of a branch-and-bound approach to further improve upon the bounds. It is obvious that by fixing some frequencies for certain links, or by limiting their domains the lower bounds are effected, and get higher. To push the lower bounds as high as presented in the following table, we had to run the branch-and-bound code for approximately 12-15 hours, on a SPARC workstation.

In the CELAR test set, problems 6, 7, 8, 9 and 10 are of the appropriate type. In problems 6, 7 and 8, however, the mobility cost is zero, and so the lower bound obtained is trivially zero. For problems 9 and 10 the results were very satisfactory. In the Table 3, we give the derived lower bounds as well as the best values known to us for these problem. The given values are the costs for assignments found by heuristics and therefore they are not necessarily optimal.

Instance	Var.	Constr.	Best known value	Lower bound
6	200	1322	3532	0
7	400	2865	344103	0
8	916	5744	276	0
9	680	4103	15665	14969
10	680	4103	32456	32144

Table 3: Lower bounding results II.

3.3.4 Branch and cut results

Feasible instances The linear programming relaxations used for feasible instances of the radio link frequency assignment problem basically consist of constraints making use of cliques in the constraint graph. Besides these, various types of valid inequalities are applied to enhance the formulation: cliques in the graph determined by the fractional solution values, interference constraints, constraints using structures that are almost cliques or related to familiar structures in facility location problems.

Since it is impractical to incorporate all constraints simultaneously and because many of them will be redundant eventually, they are generated during the process. We only add a constraint if it is sufficiently violated by the current LP solution. Some of the constraints are stored in memory, others are newly generated based on the LP solution. To prevent the LPs from becoming too large we delete constraints that are sufficiently non–binding for the LP solution. The decision when to generate constraints and when to branch is made on basis of the number of constraints added or the number of LPs solved since the previous branching. Regularly, the LP solution is used to try to generate upper bounds for the optimal value of the combinatorial problem, by trying to transform it into a feasible solution. The heuristic we use tries to assign to a link one of those frequencies for which the corresponding frequency–variable has a positive value in the LP solution. Links which have only a limited number of possible frequencies are being assigned first, in such a way that for the remaining links as much as possible potential assignments remain. The generation of such solutions enormously reduces the amount of work to be performed.

Table 4 below lists some computational results using branch and cut on the feasible RLFAP instances using the pre-processed formulations in 3.3.1. Except for CELAR05, the solution values listed are based on minimizing the number of frequencies. The values for CELAR05 are based on minimizing the largest used frequency.

The times reported are CPU times for an HP9000/720 with 144 MB of core memory, of which we used at most approximately 40MB. They do not include times for pre-processing and the generation of initial lower bounds. The table includes CELAR04 and CELAR05 for completeness although branch and cut was not used for them as they were trivial enough to be pre-processed to optimality.

The branch and cut procedure was able to find within reasonable time, feasible solutions which are at least as good as previously known solutions. More significantly, it was able to improve the previously known solution to CELAR03 and the previously known lower bound for CELAR11 thereby determining the optimal solutions of both in the process.

Instance	Initial lower bound	Found lower bound	Best value	Optimal	Approx. time (sec)
1	14	14	16		400 [1]
2	14	14	14	yes	23
3	14	14	**14**	**yes**	539
4	n.a.	46	46	yes	1 (preprocessed)
5	n.a.	792	792	yes	2 (preprocessed)
11	20	**22**	22	**yes**	6167[2]

[1]Time to generate solution of 16.

[2]Lower bound of 22 verified after 413 sec.

Table 4: Branch and cut results: feasible instances.

Infeasible instances The instances not having a feasible solution are much harder to tackle by a branch and cut framework, since it is more difficult to find linear constraints that improve the lower bound. The lack of combinatorial bounds also translates into very weak linear programming relaxations. Some constraints that have been incorporated in the framework are based on cliques in the interference graph in which the minimal distance between each of the links in the clique is (much) larger than one and the penalties of the constraints are all large. Also, we have been able to add constraints that, given we assign to a link from a certain set of frequencies, force either another link to be assigned or at least one constraint to be violated. Up to now, experiments have been performed for two test problems, which we proved to be infeasible (Table 5). The lower bounds, however, are very poor. For CELAR07 we obtained only poor solutions, since our strategies so far are designed to find lower bounds, instead of good solutions. The time required to generate the lower bounds and best value is approximately 30 minutes.

Instance	Found lower bound	Best value
6	5	3942
7	5	

Table 5: Branch and cut results: infeasible instances.

On LP-solving. During the early stages of the project, the initial branch and cut implementation incorporated a large amount of the interference constraints. We found that using an interior point method (IPM) to solve the initial linear programs was then more efficient (example 449 seconds to solve an LP using IPM versus 1939 seconds for simplex). Using constraint generation, however, solving the LPs using advanced simplex implementations by far outperforms the use of IPMs. Specifically, using preprocessing and steepest–edge pricing reduces the required computation time. We observed it often to be better to restart the primal simplex method from scratch (with the use of preprocessing), instead of using dual simplex with a warm–start.

Miscellaneous information. Before being able to apply the branch and cut algorithm, we have to generate certain inputs. This includes the search for cliques or clique–like structures in the interference graph, which is done by a greedy type algorithm and takes almost no computation time. A program was written to produce MPS–files of the initial LP relaxation in which various constraints can be taken in.

4 Recapitulation

The aim of this work element is to apply a comprehensive branch and cut strategy to solving radio link frequency assignment problems. In carrying-out this aim, we have incorporated the following components in the branch and cut framework:

1. Preprocessing to improve the initial formulation.
2. Lower bounding by generation of valid inequalities.
3. Efficient constraint management (generation and deletion).
4. Upper bounding by heuristic generation of primal feasible solutions.
5. Branching and node selection.
6. Efficient LP-solving using advanced simplex and/or interior point methods.

The techniques developed for the feasible instances proved to be effective. However, to apply these techniques effectively to the infeasible models, we have to develop more powerfull tools.

5 Conclusion

The branch and cut approach brought up several positive results, but also it has shown some drawbacks.

The major problem in solving instances of the RLFAP is that the problem formulation is much larger than the original formulation. This forces us to reduce the problem as much as possible with preprocessing techniques. Furthermore, problem decomposition by means of generating constraints and valid inequalities was effective in reducing the size of the linear programs that needed to be solved.

The lower bounding techniques provide us not only with good bounds, but also with the constraints that tighten the linear programming relaxation. The solutions of this tightened relaxation can be used effectively by primal heuristics to find good feasible solutions to the original problems. Thus, we were able to tighten the gap between lower and upper bounds, or prove optimality of the solutions generated.

With regard to the infeasible problems with penalties on the constraints, it turned out that the techniques we have tried for a branch and cut approach are still not powerful enough to generate any reasonable lower bound.

6 Perspectives

The main results indicate that the branch and cut framework can be a powerful approach in dealing with feasible instances of the RLFAP in terms of

- improving existing lower bounds (CELAR01,02,03,09,10,11),
- generating good primal feasible solutions (CELAR01,02,03,11), and
- generating optimal solutions (CELAR02,03,11)

within reasonable computational effort.

In the case of CELAR01 where the optimal is most likely strictly greater than the known/generated lower bounds, proving optimality is still a difficulty. Nevertheless, the branch and cut process is able to find good primal solutions. Generating even tighter cuts and/or devising new branching rules that might improve the generated lower bounds further is an area that can be further explored.

Clearly, much work still need to be done before the infeasible models can be tackled effectively using the branch and cut framework. Further research can be done on coming up with a more compact formulation as well as more effective pre-processing techniques and cuts for lower bounding. In addition, it may be possible to integrate the successful heuristic methods of other groups in generating primal low-violation solutions in a branch and cut framework. Overall, the results of this work element indicate that the branch and cut framework is flexible enough to allow the incorporation of a wide variety of techniques (from heuristics to exact methods) as well as problem-specific information in dealing effectively with radio link frequency assignment problems.

References

[1] K. Aardal and S. van Hoesel, "Polyhedral Techniques in Combinatorial Optimization," to appear in *Statistica Neerlandica*, 1995.

[2] R.E. Bixby, "Progress in linear programming," *ORSA Journal on Computing* **6**, 15-22, 1994.

[3] B. Borchers and J.E. Mitchell, "Using an interior point method in a branch and bound algorithm for integer programming," TR No. 195, RPI, July 1992.

[4] M. Jünger, G. Reinelt, and S. Thienel, "Practical problem solving with cutting plane algorithms for combinatorial optimization," Report No. 94.156, Institut für Informatik, Universität zu Köln, 1994.

[5] M. Jünger, G. Reinelt, and S. Thienel, "Provably good solutions for the traveling salesman problem," Preprint 92-31, Interdisziplinäres Zentrum für Wissenschaftliches Rechnen, Universität Heidelberg, 1992.

[6] N. Karmarkar "A new polynomial time algorithm for linear programming," *Combinatorica* **4**, 373-395, 1984.

[7] I.J. Lustig, R.E. Marsten, and D.F. Shanno "Interior point methods for linear programming: computational state of the art," *ORSA Journal on Computing* **6**, 1-14, 1994.

[8] J.E. Mitchell and M.J. Todd, "Solving combinatorial optimization problems using Karmarkar's algorithm," *Math. Programming* **56**, 245-284, 1992.

[9] G.L. Nemhauser, M.W.P. Savelsbergh, and G.S. Sigismondi, "MINTO, a Mixed INTeger Optimizer," *OR Letters* **15** 47-58, 1994.

[10] G.L. Nemhauser and G. Sigismondi, "A strong cutting plane/branch and bound algorithm for node packing," *J. Opl. Res. Soc.* **25** 443-457, 1992.