Hybrid algorithms for Minimum Base Problem

Zbigniew Kokosiński

Cracow University of Technology,
Faculty of Electrical and Computer Engineering,
ul.Warszawska 24, 31-155 Kraków, Poland

Summary
In this paper we present two hybrid algorithms for approximate solving of Minimum Base Problem (MBP) which is known to be NP-complete [5]. In both algorithms metaheuristic search for a solution of size k is combined with a relaxation mechanism which triggers new metaheuristic search for a solution with incremented size and another mechanism which helps to remove possible solution redundancy. Depending on the outcome of the first step of the hybrid algorithm the result may be either accepted as a final solution or used as a starting point for the second search step. The search is continued until a feasible base is found. Then, the hybrid algorithms tries to remove solution redundancy (if it exists) by using one of the two mechanisms. The new algorithms has been implemented and tested on a collection of random binary matrices as problem instances. In all cases the search space is composed of combinations of matrix columns satisfying specific problem requirements. The solver application contains SA and TS metaheuristics as its basic components. The test results are reported in terms of solution quality and computation time. A influence of the final reduction step is investigated. Some conclusions resulting from the computer experiments are derived.

Key words:
simulated annealing, tabu search, hybrid algorithm, minimum base problem

1. Introduction

In computational optimization a large number of methods and techniques was developed for finding approximate solutions of NPO problems [1], which are usually derived from known NP-complete problems [4].

Parallel and hybrid metaheuristics are considered to be interesting alternatives for exact and approximate methods developed so far [2, 8]. Many metaheuristic algorithms have been already designed and compared with existing methodologies, especially when the exact methods are not acceptable due to excessive computation time and when the approximate methods do not guarantee quality solutions, i.e. r-approximation algorithms for the problem at hand are not known. The Minimum Base optimization problem belongs definitely to such problems for which exact algorithm is often unacceptable due to huge search space and the approximate algorithms are simple heuristics which are not in APX complexity class [4]. Therefore, the metaheuristic approach seems to be well justified in this case.

In this paper we are looking for hybrid algorithms for solving optimization version of Base Problem [5].

The problem originated from logic design area, where digital circuits are modeled by Boolean matrices. For instance any combinational circuit can be modeled by input matrix $X[m,n]$ and output matrix $Y[m,p]$. Optimization problems in this area are related to circuit synthesis and decomposition in available technology: SSI, MSI, PAL, PLA, FPGA etc. One of the essential applications of algorithms for solving Minimum Base Problem is reduction of redundant arguments in input matrix X and partitioning of output matrix Y in order to enable implementation with the available logic blocks [3].

Our first attempt to built hybrid algorithms for Minimum Base Problem comes from intuition that the search space consisting of column combinations can be explored with a popular heuristic like Simulated Annealing or Tabu Search. Knowing the minimal and maximal size of the base for the given input matrix A we can start from minimal size $b = b_{min}$ as algorithm parameter and gradually increment b in consecutive metaheuristic searches till a base B of A is found. The next step of hybrid algorithm improvement resulted from the following observation: if current value of $b > b_{min}$ then the base B can be redundant. In order to detect and remove the possible redundancy the exact search is performed by generation of $(n, m-b)$-subsets of n-element set, which are candidates for reduction.

The rest of the paper is organized as follows. In the next section the decision and optimization versions of the Minimum Base Problem are defined and characterized in detail together with their know algorithms. Then, in section 3 hybrid algorithms using SA and TS algorithms as its building blocks are presented. The design assumptions and features of the developed solver are described in section 4. Testing methodology and experimental results are shown in section 5. The summary section brings conclusions and directions of future research.

Manuscript received October 5, 2020
Manuscript revised October 20, 2020
https://doi.org/10.22937/IJCSNS.2020.20.10.21

2. Minimum Base Problem

In this section formulations of BASE and MINIMUM BASE problems are given. The later will be solved by a collection of algorithms used in the experimental part of the paper.

Problem BASE

Input: binary matrix *A[m,n]* with all rows different, and a natural number *k*: $\lceil \log_2 m \rceil \leq k \leq \min\{m-1, n\}$.

Question: Is there a *k*-column base *B[m,k]* of matrix *A*, i.e. submatrix $B \subseteq A$ with all rows different?

Let us notice that *k*-column base *B* can not exist, if *(k+1)*-column base does not exist. In order to solve the decision problem BASE it is sufficient to check its sub-matrices of size *k*. Depending on the value of *k* we have different complexity cases of the problem at hand.

In general case, when $\lceil \log_2 m \rceil \leq k \leq \min\{m-1, n\}$ the exact algorithm inspects at most $\binom{n}{k}$submatrices and its complexity is exponential. In particular cases – for *m=n*, the exact algorithm inspects $\binom{n}{n-1}$ submatrices and its asymptotic complexity is *O(n)*; for *m=n–1*, the exact algorithm inspects $\binom{n}{n-2}$ submatrices and its asymptotic complexity is $O(n^2)$; for *m<n-1*, the exact algorithm inspects $\binom{n}{m-1}$ submatrices and its asymptotic complexity is $O(n^{n-m+1})$.

MINIMUM BASE PROBLEM (MBP) is the optimization version the decision problem BASE. The task is to find a minimal base *B* of the binary matrix *A*. Algorithms developed for MBP are useful in logic design for reduction of the size of the combinational circuit (CC) because enable substitution *X=A* and replacement of CC(*X,Y*) by CC(*B,Y*), where each vector $y_i \in Y$ is combinationally dependent with *B*. Another application of minimum base problem is the decomposition of one multiple-output combinatorial circuit into sub-circuits on two levels (serial-parallel decomposition).

3. Minimum Base Problem

The conducted research is based on two classical iterative algorithms: simulated annealing (SA) and tabu search (TS) that belong to the class of iterative methods [8]. Two hybrid algorithms use either SA or TS algorithms together with exact algorithm Matrix Reduction (MR) for searching solution of the Minimum Base Problem. In our research the Boolean matrices are represented by their column indices in vector representation. The neighborhood *V* is composed of vectors of the same size which differ in exactly one position. A single move is defined as a transition from the current state *S* into the next state $S' \in V$ guided by the change of the cost function *cf* of the solution. The cost function *cf* can be defined in several ways. For comparison of matrices of the same size a simple measure *ρ(B)* is be used, which denotes the number of different rows in *B*. If *B* is a base of *A* then *ρ(B)=ρ(A)*. The main cost function *cf* defined for the MBP is equal to the number of base columns.

3.1. Simulated Annealing (SA)

Simulated annealing (SA) is a well-known optimization technique described in many textbooks [8]. Markov chain is evaluated by CPU that calculates possible moves from one state to another. The key question in SA implementation is setting of algorithm's parameters like initial temperature, and a cooling schedule. For the problem at hand it is necessary to define an appropriate solution representation, cost function and a neighborhood generation scheme.

3.2. Tabu Search

Tabu search algorithm is an improvement of local search method in which so called tabu list contains a number of recent moves that must not be considered as candidates in the present iteration [8]. This feature helps the method to escape from local minima what is a main disadvantage of local search. The question is to define the solution representation, cost function, neighborhood and a single move, the size of the neighborhood and the number of candidate moves, aspiration level which decides on the possibility to accept forbidden moves if it leads to a solution improvement etc.

3.3. Hybrid algorithms

Our general hybrid algorithm can be sketched as follows:

Algorithm H_MINIMUM_BASE

1. In the first main step a classical SA/TS metaheuristic is used in searching for minimal solution of the exact size $\lceil \log_2 m \rceil \leq k \leq \min\{m-1, n\}$. If the solution is feasible (contains only different rows) it is sent to the output and hybrid algorithm terminates.
2. If this solution is infeasible then in the second step iterated executions of SA/TS take place and the constraint put on *k* is gradually relaxed until the first approximate base is met. However such a base is often redundant.
3. In the third step (the reduction phase) the solution redundancy is removed either by the next rounds of SA/TS or by the exact algorithm Matrix Reduction (MR) which searches for all possible (k–k_1)-combinations of redundant *B* columns, where: $k_1 \in \{\lceil \log_2 m \rceil, \ldots, 1\}$.

4. The solver

For all tests we used the "Minimum Base Problem Solver", which is a Python application.
The main program window contains three main fields: Matrix Generator (with "Save into a file" option) and two other fields for selected SA or TS metaheuristic. One can read or generate input data in CSV format. Numerous algorithm's parameters must be filled in the forms.
For SA it is necessary to set : initial temperature from the range [5, 10], the number of Metropolis iterations from the range [5, 25], *Alpha* coefficient from [0.6, 0.99], *Beta* coefficient from [1, 5], *MaxTime* parameter from [50, 100] and the neighborhood size $|V|$.
For TS it is necessary to set : Tabu List size *TSL* from the range [3, 8], the number of iterations *No_it* from the range [5, 25] and the neighborhood size $|V|$.

It is possible to modify metaheuristic's parameters before each main step of the hybrid algorithm.

We can record all computational results in a TXT file with any name, where all algorithms settings as well as solutions obtained after each main step of the hybrid algorithm and the corresponding computation time of each step are written. In this way we can have data of multiple runs, solution parameters and partial computational times for further statistical analysis.

5. Computational experiments

The computational experiments reported below were performed on Toshiba Satellite P75-A7200 computer, with the Intel Core i7-4700MQ CPU, 2,4 GHz and 8GB RAM, running under Windows 10 OS.
Three input matrices A : 10×10$, 20×10 and 20×20 were tested using two basic SA/TS-based hybrid algorithms and its two redundancy reduction methods.
Every time the settings of SA/TS are given, that can slightly change with the problem size (no of iterations in a single step, initial temperature for SA, size of the tabu list). The specific setting that were selected in our basic experiment are shown in Table I.

Table I Basic settings of algorithms

Matrix_A_size	*Alpha* (SA)	*No_iter* (TS)
10×10	0,8	15
20×10	0,9	20
20×20	0,95	25

The constant parameter settings for SA: *initial_temp* = 8, *no_of_Metropolis_it*=25, *Beta*=1, *MaxTime*=50, and the neighborhood size $|V|$=5.

The constant parameter settings for TS: Tabu List size *TLS*=5 and the neighborhood size $|V|$=5.
It is possible to modify metaheuristic's parameters before each main step of the hybrid algorithm.

In Tables II-IV computational data are presented. All experiments were conducted for random matrix instances generated within the application for the given sizes: 10×10, 20×10 and 20×20 - all in CSV format. The obtained Minimum Base sizes and the corresponding computation times from 10 trials are presented.
By **SA 1-2-3** and **TS 1-2-3** we denote homogeneous hybrid algorithms with all main steps performed by the selected SA/TS metaheuristic, respectively.
By **SA 1-2-MR** and **TS 1-2-MR** we denote hybrid algorithms with two main steps performed by the selected SA/TS metaheuristic and the reduction step performed by Matrix Reduction exact procedure.
We report three parameters: size of the best solution found (base size in columns), contribution of the algorithm MR in final base size reduction in columns (MR part) and computation time is [s].
Analysis of the results obtained for the three MBP instances justifies several remarks.
At first we have to state that both hybrid metaheuristics can find feasible solutions for MBP optimization problem.

The best solution quality in all cases provides **TS 1-2-MR** algorithm. The MR part significantly contributes to the final base size and without this algorithmic component TS 1-2 would achieve much worse solution for 10×10 and 20x10 problem instances than other methods. In our experiments MR component provide in average from 13.3% up to 22% reduction of the base size over SA-based methods. The dominance of **TS 1-2-MR** algorithm is confirmed by reaching optimal base size in 7/30 trials. The **TS 1-2-3** finds the optimal base in 4/30 trials. The **SA 1-2-3** finds optimum base only in 1/30 trials.

With increasing problem size even **TS 1-2** wins with **SA/TS 1-2-3** : for 20×20 problem instance TS 1-2 is defeated only by **SA 1-2-MR**. It is worth to notice that in SA-based hybrid algorithm contribution of MR part is less significant than in TS-based hybrids. This results should be validated with bigger problem instances.
Computation time of all versions for our small set if testing data does not exceed 3.7 [s]. In average case TS-based hybrid algorithms are faster than SA-based hybrids 1.9-6.1 times. The biggest ratio of computation time does not exceeds 7 (for the instance 20×20 **SA 1-2-3** time is longer 6,915 times than **TS 1-2-3** time). **TS 1-2-3** wins two times and **TS 1-2-MR** ones. The slowest method is always
SA 1-2-3.

TABLE II
COMPUTATIONAL RESULTS OF HYBRID ALGORITHMS FOR 10X10 INSTANCE OF MINIMUM BASE PROBLEM (MBP).

A matrix	SA 1-2-3		SA 1-2-MR			TS 1-2-3		TS 1-2-MR		
10x10	base size	time [s]	base size	MR part	time [s]	base size	time [s]	base size	MR part	time [s]
$\|B\| \geq 4$	8	2,673	7	0	1,539	7	0,715	8	-1	0,962
	9	2,695	7	0	1,544	7	0,727	7	-1	0,765
	8	2,676	7	0	1,575	7	0,933	8	-3	1,21
	6	2,166	9	0	2,667	7	0,836	**4**	0	0,326
	8	2,674	6	0	1,117	**4**	0,447	8	-1	0,953
	8	3,133	8	0	2,058	10	1,222	**4**	-7	1,098
	5	1,019	9	0	2,5	6	0,263	7	-1	0,409
	8	2,234	5	0	0,766	9	0,336	7	-2	0,363
	7	1,933	8	0	2,082	9	0,315	8	-1	0,421
	7	1,86	9	0	2,476	8	0,284	**4**	0	0,424
Avg=	7,4	2,306	7,5	0	1,832	7,4	**0,608**	**6,5**	**-1,7**	0,693

TABLE III
COMPUTATIONAL RESULTS OF HYBRID ALGORITHMS FOR 20X10 INSTANCE OF MINIMUM BASE PROBLEM (MBP).

A matrix	SA 1-2-3		SA 1-2-MR			TS 1-2-3		TS 1-2-MR		
20x10	base size	time [s]	base size	MR part	time [s]	base size	time [s]	base size	MR part	time [s]
$\|B\| \geq 5$	8	2,098	7	0	1,32	8	0,344	6	-2	0,31
	10	3,153	8	0	1,491	7	0,337	8	-2	0,519
	9	3,263	8	0	1,849	7	0,4	6	-2	0,303
	7	3,083	9	0	2,487	8	0,541	6	-4	0,493
	10	3,188	9	0	2,406	8	0,468	6	-4	0,449
	9	3,136	9	0	2,416	8	0,392	6	-2	0,313
	7	1,803	8	-2	2,947	9	0,419	6	-2	0,351
	7	2,15	7	0	1,342	7	0,407	8	-2	0,536
	5	0,989	8	-2	2,888	8	0,38	6	-2	0,316
	6	1,414	9	0	2,444	8	0,457	6	-3	0,398
Avg=	7,8	2,428	8,2	**-0,4**	2,159	7,8	0,415	**6,4**	**-2,5**	**0,399**

TABLE IV
COMPUTATIONAL RESULTS OF HYBRID ALGORITHMS FOR 20X20 INSTANCE OF MINIMUM BASE PROBLEM (MBP).

A matrix	SA 1-2-3		SA 1-2-MR			TS 1-2-3		TS 1-2-MR		
20x20	base size	time [s]	base size	MR part	time [s]	base size	time [s]	base size	MR part	time [s]
$\|B\| \geq 5$	7	1,686	6	-3	2,186	8	0,825	6	0	0,397
	7	2,057	6	0	0,849	**5**	0,68	**5**	0	1,469
	7	1,633	7	0	1,459	13	1,587	6	0	0,509
	8	2,115	8	0	1,906	9	1,564	6	-2	0,712
	8	3,097	8	0	1,793	11	0,482	**5**	0	0,712
	8	2,083	6	-3	2,34	9	1,167	6	-3	0,88
	9	2,552	6	-4	2,655	6	0,52	**5**	0	1,271
	7	2,037	7	0	1,289	**5**	1,472	12	-3	1,887
	10	3,665	7	0	1,282	**5**	0,53	**5**	0	0,776
	7	1,704	6	-3	2,182	6	0,531	6	-2	0,814
Avg=	7,8	2,263	6,7	**-1,3**	1,794	7,7	**0,936**	6,2	**-1**	0,943

6. Conclusions

In this paper some research results related to hybrid metaheuristics for Minimum Base Problems were reported. The conducted experiments provided a partial insight to computational behavior of hybrid metaheuristics developed on the basis of SA and TS. Some algorithms were proven to perform better than others for tested problem instances. We were interested mostly on solution quality, but computation time was also an important factor in comparison. It seems that one hybrid algorithm, i.e. **TS 1-2-MR**, is significantly better than others in terms of solution quality and computation time. We believe that the presented research results are encouraging enough to foster further experiments with our solver for more complex input instances.

Acknowledgments

The presented research was conducted at the Faculty of Electrical and Computer Engineering, Cracow University of Technology and was financially supported by Ministry of Science and Higher Education, Republic of Poland.

The Python application was coded by Mr. Dawid Leszczyk, who implemented the hybrid algorithm as a part of his B.E. project completed in February, 2020.

References

[1] G. Ausiello. et all., Complexity and Approximation Combinatorial optimization problems and their approximability properties, Springer-Verlag, 1999.
[2] C. Blum, A. Roli, E. Alba, An Introduction to Metaheuristic Techniques, [in:] Parallel Metaheuristics, Wiley-Interscience, 3-42, 2005.
[3] G. De Micheli, D.S. Johnson, Synthesis and optimization of digital circuits, McGrawHill, 1994.
[4] R. Garey, D.S. Johnson, Computers and intractability. A guide to the theory of NP-completeness, Freeman, San Francisco, 1979.
[5] A. Kapralski, W. Skarbek, Problem of searching minimum base included in Boolean tables, Podstawy Sterowania, Vol.16, No.3-4, 257-265, 1986.
[6] M. Laguna M., J. Wesley, F.W. Glover, Tabu search methods for a single machine scheduling problem, 1989.
[7] S. Łukasik, Z. Kokosiński, G. Świętoń, Parallel simulated annealing algorithm for graph coloring problem, PPAM'2007, Lect. Notes in Comp. Sc., Vol. 4967, 229-238, 2008.
[8] S.M. Sait, H. Youssef, Iterative computer algorithms with applications in engineering.

Zbigniew Kokosiński received his M.S. degree in 1982 from Cracow University of Technology, Kraków, Poland. In 1992 and 2019 he received Ph.D. degree with distinction and D.Sc. degree, respectively, both in Computer Science from the Gdańsk University of Technology, Gdańsk, Poland. In 1994-1997 he was employed at the Department of Computer Software, University of Aizu, Aizu-Wakamatsu, Japan. Currently, Z. Kokosiński is a Professor at the Dept. of Automatic Control and Information Technology, Faculty of Electrical and Computer Engineering, Cracow University of Technology. His research is focused on combinatorial optimization and parallel metaheuristics, generation of combinatorial objects in parallel, associative processors and algorithms, programmable devices and systems. Prof. Kokosiński published over 50 refered papers in scientific journals and conference proceedings, chapters in books, and also was granted 2 patents for digital circuitry design. Member of Editorial Board of International J. of Computing. Reviewer for Information Sciences, J. of Parallel and Distributed Computing, The Computer J., Parallel Computing, Parallel Processing Letters, Networks, Optimization, Lecture Notes in Computer Science, ACM J. of Experimental Algorithmics, J. of Circuits, Systems and Computers, J. of Mathematical Modeling and Algorithms, Kragujevac J. of Mathematics, Studia Informatica, J. of Telecommunications and Information Technology etc. Prof. Kokosiński participated in organization of many international conferences in the area of computing as a program committee member, referee, session chair etc. (PDCS, ICEC, IPPS, PDPTA, PPAM, FPL, PARELEC, MCCSIS, ISPDC, ICACIT, ICAISC, FedCSIS, Future Computing, iCAST, SCET, IDAACS). His professional memberships included: IPSJ, IEEE CS, ACM and IASTED.
Biographical notes: Marquis "Who's Who in Science and Engineering" (2007).