Hybridizing Reactive Tabu Search with Simulated Annealing

Stefan Voß[1] and Andreas Fink[2]

[1] Institute of Information Systems (IWI), University of Hamburg
Von-Melle-Park 5, 20146 Hamburg, Germany
stefan.voss@uni-hamburg.de
[2] Chair of Information Systems, Helmut-Schmidt-University / UniBw Hamburg,
Holstenhofweg 85, 22043 Hamburg, Germany
andreas.fink@hsu-hamburg.de

Abstract. Reactive tabu search (RTS) aims at the automatic adaptation of the tabu list length. The idea is to increase the tabu list length when the tabu memory indicates that the search is revisiting formerly traversed solutions. Once too many repetitions are encountered, an escape mechanism constituting a random walk is an essential part of the method. We propose to replace this random walk by a controlled simulated annealing (SA). Excellent results are presented for various combinatorial optimization problems.

1 Introduction

The basic paradigm of tabu search is to use information about the search history to guide local search approaches to overcome local optimality. In general, this is done by a dynamic transformation of the local neighborhood. RTS aims at the automatic adaptation of the tabu list length [1, 2]. A possible specification can be described as follows: Starting with a tabu list length s of 1, it is increased to $\min\{\max\{s+2, s \times 1.2\}, b_u\}$ every time a solution is repeated, taking into account an appropriate upper bound b_u (to guarantee at least one admissible move). If there has been no repetition for some iterations, we decrease it to $\max\{\min\{s-2, s/1.2\}, 1\}$. To accomplish detecting repetitions of solutions, we apply a trajectory based memory using hash codes.

For RTS it is appropriate to include means for diversifying moves whenever the tabu memory indicates that we are trapped in a certain region of the search space. As a trigger mechanism one may use, e.g., the combination of at least three solutions each having been traversed three times. The standard escape strategy is to perform randomly a number of moves (depending on the average of the number of iterations between solution repetitions) [1, 2]. As termination criterion one may consider a given time limit. In this paper we propose to replace this random walk by a controlled SA.

The next section provides details of the specific hybridization that we propose. In Section 3 we sketch the set of problems that we have currently looked at. The paper closes with some conclusions.

2 The Hybrid Method

SA extends basic local search by allowing moves to worse solutions. Starting from an initial solution, successively a candidate move is randomly selected; this move is accepted if it leads to a solution with a better objective function value than the current solution, otherwise the move is accepted with a probability that depends on the deterioration Δ of the objective function value. The acceptance probability is computed according to the Boltzmann function as $e^{-\Delta/T}$, using a temperature T as control parameter. Various authors describe robust realizations of this general SA concept. Following [4], the value of T is initially high, which allows many worse moves to be accepted, and is gradually reduced through multiplication by a parameter *coolingFactor* according to a geometric cooling schedule.

Instead of using random walk as the escape mechanism within RTS, we propose to apply SA, which performs, depending upon the parameter setting, diversification as well as intensification to some degree. In the computational experiments described in this paper, we examine the effect of adapting the SA parameter values in accordance with its primary role as diversification mechanism [4]. We stick to using $\alpha = 0.95$, whereas *frozenLimit* is set to 1 in order to terminate earlier. Instead of $\mathit{initialAcceptanceFraction} = 0.4$ we also use the value 0.1 which means less diversification; instead of $\mathit{sizeFactor} = 16$ we also use the value 1 which speeds up the cooling process; instead of $\mathit{frozenAcceptanceFraction} = 0.02$ we also use the value 0.1 which eventually means less intensification. (Whenever a SA run is performed while an overall time limit is reached we finish that run before terminating the approach.)

3 Computational Results

We have considered various optimization problems to emphasize the impact of the RTS/SA-hybridization proposed above. In the sequel we provide results for the *Ring Load Balancing Problem* (RLB) and mention other problems where implementations and results are available. All implementations have been performed by using our HOTFRAME software [3] on an average PC.

3.1 Ring Load Balancing Problem

The RLB is an NP-hard telecommunications problem where we are given a ring of nodes with a set of communication demands between node pairs [5]. Assuming that the communication demands occur simultaneously, the task is to decide for each demand whether to route it clockwise or counterclockwise, minimizing the maximum bandwidth requirement on any of the links between adjacent nodes. That is, given a set of n nodes and a set of demands between pairs of nodes, find a direction for each of the demands so that the maximum of the loads on the links in the network is as small as possible.

The solution space consists of all possible routing directions for the demands. We employ a straightforward neighborhood that is defined by switching the routing direction for one demand (node pair). The quality of such a local search move is assessed by the implied change of the objective function value. The RLB has been used as a testbed as optimal solutions are available and we are yet able to show the impact of our approach. We report results for problem instances of the RLB proposed in [5].

Table 1 provides a detailed view on the characteristics of the data. In the first column we describe the different scenarios together with (n, D), the number of nodes n and number of demands D. The first three blocks are non-centralized demands while the last block gives centralized demands, i.e., $D = n - 1$. Each row refers to an average of ten runs. Correspondingly, column 'opt' provides (in each row) the average of the optimal solution values for these ten runs. We provide results for the case where diversification is performed by using the original random walk as an escape mechanism (Esc.=RW) with time limit 1 second. On the right side of the table we consider the case where the escape mechanism is performed by applying a SA run with the standard parameter setting as described above. The time limit is the same as before as well as one with a possible instance-dependent extension based on the given data. While all approaches provide small deviations from optimality the hybrid approach is able to considerably improve on its pure counterpart.

3.2 Other Problems

Additional problems where we have applied our ideas include, among others, the *Minimum Weight Vertex Cover Problem* and the *Minimum Labelling Spanning Tree Problem*. For all considered problems we show that the hybridization improves the numerical results of the pure RTS and the SA.

4 Conclusions

In this paper, we have presented a very simple and yet very effective modification of the well-known reactive tabu search. As a conclusion we may deduce that randomness helps in metaheuristics, though a controlled way of incorporating randomness might be more successful than pure randomness. The number of successful implementations of RTS in literature provides an option to revisit those implementations to crosscheck whether our idea also holds in those applications that have not been looked at in this paper.

References

1. R. Battiti. Reactive search: Toward self-tuning heuristics. In V.J. Rayward-Smith, I.H. Osman, C.R. Reeves, and G.D. Smith, editors, *Modern Heuristic Search Methods*, pages 61–83. Chichester: Wiley, 1996.

		opt	Esc.=RW (1s)	Esc.=SA (1s)	Esc.=SA (max(1,D/90)s)
ss	(5;6)	131.5	131.5	131.5	131.5
	(10;12)	231.6	231.6	231.6	231.6
	(15;25)	507.5	507.5	507.5	507.5
	(20;40)	734.5	734.5	734.5	734.5
	(25;60)	1013.4	1013.3	1013.3	1013.3
	(30;90)	1435.8	1434.9	1434.2	1434.2
mm	(5;8)	173.6	173.6	173.6	173.6
	(10;23)	422.5	422.5	422.5	422.5
	(15;50)	883.8	882.2	882.2	882.2
	(20;95)	1457.5	1457.4	1455.8	1455.8
	(25;150)	2253.3	2241.0	2234.3	2234.2
	(30;200)	3013.2	3019.1	3006.8	3006.8
ll	(5;10)	186.0	186.0	186.0	186.0
	(10;45)	728.6	728.3	728.3	728.3
	(15;105)	1605.1	1602.2	1599.9	1599.9
	(20;190)	2742.3	2736.2	2721.0	2720.6
	(25;300)	4243.5	4238.7	4225.2	4221.3
	(30;435)	5982.0	5987.7	5968.2	5956.9
ce	(5)	155.0	155.0	155.0	155.0
	(10)	349.4	349.4	349.4	349.4
	(15)	530.7	530.7	530.7	530.7
	(20)	721.8	721.8	721.8	721.8
	(25)	991.8	991.8	991.8	991.8
	(30)	1101.7	1101.7	1101.7	1101.7
		Average:	0.12%	0.03%	0.01%

Table 1. Computational results for the RLB.

2. R. Battiti and G. Tecchiolli. The reactive tabu search. *ORSA Journal on Computing*, 6:126–140, 1994.
3. A. Fink and S. Voß. HotFrame: A heuristic optimization framework. In S. Voß and D.L. Woodruff, editors, *Optimization Software Class Libraries*, pages 81–154. 2002.
4. D. S. Johnson, C. R. Aragon, L. A. McGeoch, and C. Schevon. Optimization by simulated annealing: An experimental evaluation; part I, graph partitioning. *Operations Research*, 37:865–892, 1989.
5. Y.-S. Myung, H.-G. Kim, and D.-W. Tcha. Optimal load balancing on SONET bidirectional rings. *Operations Research*, 45:148–152, 1997.