

Title: Heuristic and Metaheuristic Approaches for Optimizing Traffic Signal Control

Topic: Optimization Technologies

Author: Michael Shenoda, P.E., Ph.D.

Department: Department of Civil Engineering, The College of New Jersey

Mailing Address: PO Box 7718
Ewing, NJ 08628

E-mail: shenoda@tcnj.edu

Phone: (609) 771-3017

Fax: (609) 637-5135

Abstract

Vehicles utilizing a signalized intersection can have their travel performance considered on the basis of their stopped delay at that intersection. However, the optimization of the traffic signal's operation based on this measure of effectiveness, as well as others, is complicated as the number of vehicles arriving and the number of phases for the signal. These factors can create an extremely large solution space for the optimization. In performing this optimization for a signalized intersection, the computation time is key to the success of implementation of the solution in the short term (e.g. from phase to phase or cycle to cycle). This proposal discusses the development of a heuristic approach, via Webster's method for signal splitting, and a metaheuristic approach, via tabu search, to finding the solution to the signalized intersection optimization problem. The approaches devised herein were found, by experimentation over several hundred runs, to reduce the computation times by an average of 44% and by up to 76%, in some cases, as compared to an exhaustive search of the solution space. They were also found to come within an average of 10% of the optimal solution and reduce the variance in the final solution found from run to run by an average of 33%. These results would allow improved feasibility for implementation of more complex traffic control strategies with greater impacts on the reduction of congestion at signalized intersections.

Introduction/Background

When vehicles utilize a signalized traffic intersection, the performance of that intersection can be measured objectively. This is done through a variety of measures of effectiveness (MOEs) which can include [via Pignataro (1973) and May (1990)]:

- the number of stops made by vehicles
- the time vehicles are required to be stopped, or stopped delay (total or average per vehicle)
- the time vehicles spend in the system

- travel time
- the difference between the travel time and the minimum time that vehicles can spend in the system, or total delay

Traffic control systems have been devised over the past 40 or so years that have begun to take advantage of the ability to more easily and readily measure these MOEs, whether through the use of detectors, video equipment, or RFID (e.g. toll transponders). These systems, if they can respond directly to changes in these MOEs with changes in the traffic stream, are termed adaptive traffic signal systems. The most prominent in terms of use and application are:

- Optimized Policies for Adaptive Control (OPAC) [Gartner (1983)]
- Split Cycle Offset Optimization Technique (SCOOT) [Greenough and Kelman (1998)]
- Sydney Coordinated Adaptive Traffic System (SCATS) [Lowrie 2001]
- Real-time Hierarchical, Optimized, Distributed, Effective System (RHODES) [Mirchandani and Head (2001)]

These all use, to some degree, a means of evaluating traffic streams in either recently-past or upcoming short-term time periods (ranging from a few seconds to five minutes in length). Their evaluations are either exhaustive in nature over basic criteria (e.g. presence or absence of vehicles in a given time period), or examine time periods with fixed lengths (e.g. a set time lag between intersections, or an offset). However, these methods do not utilize the full potential of data collection systems to provide information about traffic streams. More recently, adaptive signal control methodologies in development utilize the arrival and departure data regarding individual vehicles to determine performance via MOEs.; these include Cai et. al. (2009) and Fang and Elefteriadou (2010). Such systems can operate over time periods based on the travel patterns of the vehicles, so that neither the time period lengths nor positions are fixed over an evaluation horizon. The traffic signals in such systems are controlled by the optimization of an MOE that can be calculated directly from this arrival and/or departure data.

Herein, the system devised by Shenoda and Machemehl (2006) will be considered for optimization. In it, the basic concept is that stopped delay is calculated based on the difference between each vehicle's arrival timepoint and the timepoint at which the traffic signal changes to red. The timepoint is then shifted until the sum of the vehicles' stopped delays is minimized. An examination of the formulation and sample situations and configurations shows that the algorithm involves a non-convex optimization.

Consideration of the use of heuristics

Even within a limited evaluation horizon, such an optimization would seem to have an infinite number of possible solutions, and thus an efficient search procedure would lend itself ably to the task. Finding the optimum timepoint would not be guaranteed, but given enough operating time and a suitable method, a solution within some acceptable margin of error of the optimum could be reached. Numerical search methods over non-convex intervals generally involve using a one-dimensional search (e.g. steepest descent, golden section, etc.) of the range over unimodal intervals. The specifics of these types of methods will not be discussed herein, but are readily referenced through literature on optimization; Fletcher (1987) is a good reference for the specifics of these methods. There are drawbacks, however, to the use of these in that, depending

on the acceptable error and the initial solution used, they can be time-consuming and may prevent reaching an acceptable solution in a practical time frame. Other limiting factors in using this type of approach are the time step over which the optimization is iterated and the number of phases over which the system needs to be optimized, which can expand the computation time exponentially.

In light of this, one might consider that there may not be an infinite number of possible timepoints, but that this number is practically limited. It is simplest to consider that the timepoints, in partitioning the horizon into “green signal” and “red signal” portions, should have a limiting resolution based on the system performance. One end of the spectrum of resolutions might be, for instance, the transmission speed between the processor and the signal controller, which may be on the order of 10^{-3} seconds or lower [AASHTO et al. (2004)], and the other might be the minimum headway between vehicles at freeflow speed, which is generally assumed, based on Greenshields et al. (1947), to be approximately 2 seconds. This puts the number of practically possible timepoints in the range of less than 10^2 to 10^4 per phase.

When we impose this limit to the number of timepoints, the possibility of exhaustive evaluation might re-enter into consideration. A potential drawback to the limitation of timepoints with this approach, however, is that, if the resolution is not small enough, it may not be possible to capture the time difference between arrivals of vehicles at multiple approaches. This may eliminate the ability to utilize potential timepoints that could provide delay improvement. (This is, in fact, an additional drawback to numerical search methods, which can exhibit this fault if the acceptable error in the timepoint solution is not small enough.) Another drawback is, as with numerical search methods, that as the intersection to be optimized become more complex, so does the computation time.

If one considers the manner in which delay is calculated, based upon vehicle arrivals, one may actually see that delay can be optimized most readily by making switches in the signal states only within some interval of these critical points along the horizon. This is to say that delay can only approach a minimum by switching the signal state as quickly as possible after a vehicle passes through an approach that is presently served by a green signal. When considering this, the problem becomes more like a combinatorial optimization problem. In it, the timepoint for which to change the signal state determines the placement of a vehicle into one of three sets: a “red before green” set, a “green” set and a “red after green” set. The delay for each vehicle is a set value for each of the now-limited timepoints to be considered; the determination then becomes one of which timepoints will allow which delay values to be considered in which set, determining whether the delay value in question will contribute to the overall delay.

It is also possible, with the number of possible timepoints reduced to, essentially, the number of vehicles arriving at the intersection during the evaluation horizon, to perform an exhaustive enumeration of the potential timepoints to find the optimal solution. It may also be possible to construct a numerical search procedure through an integer programming approach after this reduction. The overall number of combinations using one of these approaches would be on the order of the maximum number of vehicles wishing to utilize a phase during a given horizon to the power of the number of phases. However, as the number of vehicles and the number of phases increase, the computation time for an exhaustive enumeration would increase exponentially. For instance, on a 4-phase intersection where a maximum of 30 vehicles need to

be processed for a phase, the number of possible timepoint combinations is 30^4 , or approximately 810,000 combinations. This type of problem could be easily encountered where a phase processes more than one approach (e.g. one phase for both directions of a street).

The “Proportional Heuristic”

A basic heuristic, or problem-solving approach, may be applied to the problem. The goal of this heuristic is limited, in that it will only do the following:

- a) provide a useful starting point for optimization
- b) provide a baseline value by which to rank successive timepoints or combinations thereof to evaluate

The evaluation can then be truncated based on the time constraints of the traffic signal, as discussed earlier.

In this case, a solution will be used based on what can be considered a “proportional heuristic”. The concept requires starting with a “proportional timepoint” for each phase, calculated using the proportion of vehicles wishing to utilize that phase during a horizon to the total number of vehicles arriving at the intersection during that horizon, much like Webster’s method for resolving phase lengths in a cycle (Pignataro (1973) and May (1990)). Using this set of proportional timepoints, an initial solution is obtained, with the timepoint, λ , for each phase k that has the smallest deviation from the proportional timepoint for that phase. This can be considered over the set of timepoints for a phase generated by each vehicle j (i.e. λ_{kj})
Mathematically, this initial solution is:

$$\lambda_k = \lambda_{kj} \text{ such that } \min[\lambda_{kj} - (\# \text{ of veh. for } k)/(\text{total } \# \text{ of veh.})]$$

In order to fully utilize the proportional heuristic, the potential combinations timepoints must be exhaustively enumerated, and then ranked for sequential use. This ranking is done by ascending order of the sum of the differences between the proportional timepoint and the λ_k value in the combination for each phase. Therefore, solutions which deviate less from the proportional lambda set will be ranked higher and available for use in the algorithm at earlier iterations. Using the solutions based on this ranking system, the algorithm should be able to arrive at a good, if not optimal, solution, much more quickly than ordinal exhaustive enumeration. It should be noted that this situation is not guaranteed, as peculiarities in the arrival time set (e.g clustering arrivals at the extremes of the horizon) may produce an optimum timepoint far from the proportional lambda for a particular phase. Furthermore, use of the proportional heuristic is only preferable on smaller-scale problems, as the process of applying the heuristic to completely solve the algorithm currently requires exhaustively ranking the solutions and then exhaustively searching them; as the number of solutions grows, the process eventually takes more computational time than is saved in reaching the optimal, or a good, solution in an earlier iteration.

Potential benefits of a metaheuristic approach

Clearly, a basic heuristic such as the one outlined above is quite useful in some small-scale cases, but may not be as useful in others. There are also situations where, even in a truncated search where a good solution might be expected after a limited number of iterations, the heuristic above

may be problematic. These would involve the “platooning”, or clustering, of vehicles at various time positions within the evaluation horizon. It is possible for a small platoon of vehicles to shift the optimal timepoint by a distance outside to its size relative to the overall traffic stream, simply by dint of its expansive delays. This would render the proportional value to be no more suitable a criterion for consideration than any other timepoint. The heuristic may be adjusted for intersections where such platoon-based behavior is anticipated, but this would not be in the spirit of the “adaptiveness” of a control methodology. A heuristic that allows a shift in the evaluation criterion based on prior patterns of arrival may be a basis for further research.

There may be other drawbacks to the use of a basic heuristic for the adaptive control scenario. A situation where there is a high demand regime to be processed by one phase and lower demand to be processed by the others would create many combinations where potential timepoint values of zero hold the places in the solution space where vehicle arrival times in those phases are nonexistent. Possibilities to streamline a search that involves this issue would be the creation of an alternative representation of the solution space (e.g. forward or backward star) to eliminate the zero placeholders and reduce the number of combinations. However, without excluding this as a possibility for further exploration, especially for less complex applications, the benefits of such an approach would only be significant in a highly disparate demand situation. This is not guaranteed to be present, from phase to phase, even for the same intersection. In any demand distribution case, the increase in complexity and overall vehicular demands on intersections to be addressed by the control methodology would likely outstrip the benefits of such an approach. Furthermore, this increase is only considered on a single-intersection basis; operation on an arterial or network basis can render exhaustive enumeration quite cumbersome. A major constraint, the availability of computation time based on the time step of the iterations and the “real-time” demands of the algorithm’s operation, leads to the consideration of other time-saving alternatives.

Application of the metaheuristic approach

Tabu search is a metaheuristic search procedure that addresses many of the pertinent issues while providing a clear logic to the transition between possible timepoint combinations to be considered. A discussion of tabu search can be found in Glover (1989), but the process can be summarized as follows:

1. An initial solution is generated and its objective value is taken as the current best value.
2. A neighbor (or a candidate, from a list more restrictive than the neighborhood) to the initial solution is selected based on a move.
3. Its objective value is determined:
 - a. If the neighbor’s objective value is less than that of the initial solution, then that value becomes the current best value.
 - b. If the neighbor’s objective value is greater than that of the initial solution, then some attribute of the move becomes a tabu restriction of any move (i.e. a move to a neighbor involving that attribute cannot be made) for some number of iterations (the tabu tenure).
4. If 3(a) is true, then a non-tabu move is made from the initial solution; if 3(b) is true, a non-tabu move is made from the neighbor.

5. Non-tabu moves can be made until:
 - a. the neighborhood or candidate list is exhausted.
 - b. an aspiration criterion (one that allows a move that is considered tabu to be made) is applied.
 - c. a stopping criterion (e.g. end of available computation time, minimal improvement in solution, etc.) is met.

Tabu search has been applied to a wide variety of operations research problems, and has had a number of enhancements made to it which have allowed it to tackle rather complex problems in this area. A rather simple tabu search application was used as a framework for the adaptive traffic signal control optimization in its current setup; this is similar to that of Laguna et al. (1990) for the single machine scheduling problem, which essentially involves the basic process described above.

The search can start with the array of timepoint values, which has a number of rows equal to the maximum number of vehicles arriving at a phase and a number of columns equal to the number of phases to be considered. A timepoint is selected for each k , or phase, in the array. Then, a neighborhood of that solution could have all the timepoint values held except for one, which could vary based on a move up and down a particular column. Such a move would be defined as a low influence move. The neighborhood could also allow the columns to switch for that set of timepoint values, so that the green times could be allowed to occur in a different order during the horizon. This would be defined as a large influence move. The current implementation only uses low influence moves, adhering to the constraint of increasing sequential timepoint values. In the current case, the set of proportional heuristic solutions is generated (as described below) and chosen from for the initial solution. Based on the occurrence of a non-decreasing delay, a set of timepoint values has one of its elements assigned as tabu. The element and its location in the timepoint set are placed in arrays where they remain for the length of the tabu tenure. If a timepoint set is then selected with the tabu element in the stored location before the tabu tenure is completed, that timepoint set is rejected and another timepoint set is leaped to in the proportional solution space, which is the neighborhood in this particular application. The tabu array can hold as many element-location pairs as the value of the tabu tenure.

The aspiration criteria have a large effect on the ability to explore solutions, since it is quite possible that a tabu or non-neighborhood move may lead to a significantly improved solution compared to the current best solution. The two main applicable aspiration criteria are these:

- an “aspiration by default” criterion, where, if all moves are tabu, a move will be allowed that led to the objective value closest to the current best value
- an “aspiration by influence” criterion, where, if a “high influence” move has been performed since an attribute became tabu, “low influence” moves involving that attribute are now allowed

A version of the “aspiration by default” criterion is used herein, whereby, if all moves are tabu, the selection that was originally next in the proportional solution space is selected.

Testing of the approaches

The proportional heuristic and the metaheuristic approach described above were coded in C++ and embedded into the adaptive traffic control methodology outlined in Shenoda and Machemehl (2006). An intersection was configured to test the effectiveness of the various techniques for evaluation. The tested intersection had four legs and an independent phase for each approach. The first four runs were conducted with varying traffic demands moving straight only on each leg. The second four runs were conducted adding a fifth “approach”, with traffic turning left from one of the legs, having its own fifth phase.

The methodology was allowed to operate for 3600 seconds for each run, and the following data were recorded:

- number of recursions (optimizations) performed over the run time
- average iteration at which the lowest stopped delay was reached for the run
- standard deviation of delays over the course of each set of iterations
- overall stopped delay (in vehicle-seconds) accumulated during the run

These values are tabulated in Table 1.

platform	attribute	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8
Exhaustive	avg. iteration #	178	247	35	116	44	1011	154	40
	# of recursions	160	162	262	191	265	190	427	271
	std. deviation	126.2	159.9	34.8	93.1	75.4	1493.2	257.5	51.4
	stopped delay	6766	8223	3705	5841	2938	6544	3632	2970
Proportional	avg. iteration #	74	78	26	63	n/a	n/a	n/a	n/a
	# of recursions	167	179	260	200	n/a	n/a	n/a	n/a
	std. deviation	23.1	19.9	26.3	31.6	n/a	n/a	n/a	n/a
	stopped delay	6416	7377	3758	5654	n/a	n/a	n/a	n/a
Metaheuristic	avg. iteration #	102	146	29	81	128	240	104	106
	# recursions	161	159	267	191	270	282	451	274
	std. deviation	65.1	95.4	42.1	73.9	166.5	137.3	122.3	152.9
	stopped delay	6773	7816	3463	6042	2739	4798	3334	3094

Table 1. Results of approach testing on a sample intersection.

These results demonstrate the following benefits of application of the proportional heuristic and or the metaheuristic approach:

- general reduction in number of iterations required to reach lowest delay
- delay values within reasonable deviation from the exhaustive search values
- general reduction in the standard deviation of the solutions within each run

The numerical results are also summarized in Table 2.

critierion	platform	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Avg.
reduction in number of iterations	proportional	58.4%	68.4%	25.7%	45.7%					49.6%
	metaheuristic	42.7%	40.9%	17.1%	30.2%	-190.9%	76.3%	32.5%	-165.0%	-14.5%
deviation from exhaustive delay	proportional	-5.5%	-11.5%	1.4%	-3.3%					-4.7%
	metaheuristic	0.1%	-5.2%	-7.0%	3.3%	-7.2%	-36.4%	-8.9%	4.0%	-7.2%
reduction in variance	proportional	57.3%	64.7%	13.1%	41.7%					44.2%
	metaheuristic	28.2%	22.8%	-10.1%	10.9%	-48.6%	69.7%	31.1%	-72.5%	3.9%

Table 2. Summary of results of heuristic and metaheuristic approach testing.

While it is noted that the averages do not always reflect a strong demonstration of the points above, it is clear that, for a vast majority of the runs, the proportional heuristic and the metaheuristic approach outperform exhaustive evaluation for the testing.

Conclusion

Use of heuristic and metaheuristic approaches show clear potential to reduce the computation time for the large solution spaces involved in optimizing traffic signal control. They can also “tighten up” searches over a large solution space in a relatively short computation time. Such benefits are crucial in allowing the functioning of intricate algorithms for traffic signal control within the time constraints required for both equipment operation and changes in the traffic stream. There are also valuable in accommodating the development of adaptive traffic signal control to incorporate more accurate and comprehensive traffic data. Further research will allow such heuristic and metaheuristic approaches to be even more robust and viable in the traffic signal control milieu.

References

AASHTO, ITE and NEMA. Advanced Transportation Controller (ATC) Standard for the Type 2070 Controller, Version 2.03, March 12, 2004. American Association of State Highway and Transportation Officials, Institute of Transportation Engineers, and National Electrical Manufacturers Association. http://ite.org/standards/atc/ATC2070_2.03.pdf. Accessed July 1, 2005.

Cai, C., C.K. Wong, and B.G. Heydecker. Adaptive traffic signal control using approximate dynamic programming. *Transportation Research Part C: Emerging Technologies* Volume 17, Issue 5. TRB, National Research Council, Washington, DC, 2009, pp. 456-474.

Fang, F.C. and L. Elefteriadou. Modeling and simulation of vehicle projection arrival–discharge process in adaptive traffic signal controls. *Journal of Advanced Transportation* Volume 44, Issue 3. John Wiley & Sons., Inc., New York, 2010 pp. 176–192

Fletcher, R. *Practical Methods of Optimization*. Wiley, New York, 1987.

Gartner, N.H. OPAC: A Demand-Responsive Strategy for Traffic Signal Control. In

Transportation Research Record: Journal of the Transportation Research Board, No. 906, TRB, National Research Council, Washington, DC, 1983, pp. 75-81.

Glover, F. Tabu Search – Part 1. In ORSA Journal on Computing, Vol. 1, No. 3. Operation Research Society of America, 1989.

Greenough, J.C. and W.L. Kelman. Metro Toronto SCOOT: Traffic Adaptive Control Operation. ITE Journal, Vol. 58, No. 5, May 1998.

Greenshields, B.D., Shapiro, D. and Ericksen, E.L. Traffic Performance at Urban Intersections. Technique Report No. 1. Bureau of Highway Traffic, Yale University, New Haven, CT, 1947.

Laguna, M., J.W. Barnes and F. Glover. Tabu search methods for a single machine scheduling problem. In the Journal of Intelligent Manufacturing, Vol. 2, pp. 63-74. Chapman and Hall, 1991.

Lowrie, P. SCATS – The History of Its Development. Presented at the 8th World Congress on Intelligent Transportation Systems, Sydney, Australia, 2001.

May, A.D. *Traffic Flow Fundamentals*. Prentice Hall, Englewood Cliffs, NJ, 1990.

Mirchandani, P. and L. Head. A Real-Time Traffic Signal Control System: Architecture, Algorithms and Analysis. In Transportation Research Part C Vol. 9, No. 6. TRB, National Research Council, Washington, DC, 2001, pp. 415-432.

Pignataro, L.J. *Traffic Engineering: Theory and Practice*. Prentice Hall, Englewood Cliffs, NJ, 1973.

Shenoda, M. and R. Machemehl. Development of a Phase-by-Phase, Arrival-Based, Delay-Optimized Adaptive Traffic Signal Control Methodology with Metaheuristic Search. Technical Report Number 167863-1, Southwest Region University Transportation Center, College Station, TX, October 2006.