# The Constrained Shortest Path Problem: Algorithmic Approaches and an Algebraic Study with Generalization*

Ying Xiao[1], Krishnaiyan Thulasiraman[1], Guoliang Xue[2] and Alpár Jüttner[3]

1. School of Computer Science, University of Oklahoma, Norman, USA
2. Dept. of Computer Science and Engineering, Arizona State University, Tempe, USA
3. Dept. of Operations Research, Eötvös University, Budapest and the Ericsson Traffic Laboratory, Hungary

**Abstract:** The constrained shortest path (CSP) problem requires the determination of a minimum cost *s*-*t* path with delay at most a nonzero integer *T*. In this paper, we first point out the equivalence of certain algorithms, simply called the LARAC (Lagrangian Relaxation Based Aggregated Cost) algorithm presented independently in some earlier works. The LARAC algorithm solves the integer relaxation of the CSP problem (RELAX-CSP) and is based on a geometric approach. We then present an algebraic study of RELAX-CSP and establish several new properties of the optimal solution. These properties also hold for general combinatorial optimization problems involving two additive parameters. We follow this by establishing a characterization of optimal solutions for the general CSP problem involving more than two additive parameters. We present a new heuristic called LARAC-BIN based on binary search. This heuristic involves a parameter whose value can be specified in advance depending on the allowable deviation of the cost from the optimum. Using Megiddo's parametric search, we also present a strongly polynomial time algorithm for RELAX-CSP. This algorithm has the best complexity to date for RELAX-CSP. Finally, we present an integrated approach to the CSP problem and show how the LARAC algorithm can be used to achieve considerable speedup of ε-approximation algorithms for the CSP problem.

**Keywords:** Constrained shortest path problem, discrete optimization, approximation algorithm, heuristic approaches.

---

* The work of Krishnaiyan Thulasiraman was supported in part by NSF ITR grant ANI-0312435. The work of Guoliang Xue was supported in part by NSF ITR grant ANI-0312635 and ARO grant DAAD19-00-1-0377.

# I. Introduction

Shortest path and minimum cost flow/ maximum flow computations are fundamental problems in operations research. Though interesting in their own right, algorithms for these problems also serve as building blocks in the design of algorithms for complex problems encountered in large scale industrial applications. So, over the years there has been an extensive literature on various aspects of these two problems. Both these problems are solvable in polynomial time. But adding one or more additional additive constraints makes these problems intractable. In this paper, we focus on the constrained shortest path (CSP) problem. This problem requires determination of a minimum cost path from a source node to a destination node of a network subject to the condition that the total delay of the path be less than or equal to a specified value. We shall also consider certain aspects of the problem when the minimum cost path is required to satisfy more than one additive constraint.

The constrained shortest path problem has attracted considerable attention from different research communities: operations research, computer science, and telecommunications. The interest from the telecommunications community arises from the great deal of emphasis on the need to design communication protocols that deliver certain performance guarantees. This need, in turn, is the result of an explosive growth in high bandwidth real time applications that require stringent QoS guarantees. It is for this reason that the CSP problem has assumed great importance in telecommunication network applications.

It has been shown in [24] that the CSP problem is NP-complete even for acyclic networks. So, in the literature, heuristic approaches and approximate algorithms have been proposed. Heuristics, in general, do not provide performance guarantees on the quality of the solution produced, though they are usually fast in practice. On the other hand, $\varepsilon$-approximation algorithms deliver solutions with cost within $(1 + \varepsilon)$ time the optimal cost, but are usually very slow in practice because they guarantee the quality of the solutions produced.

Approximation algorithms for CSP problem are usually based on scaling and rounding of data. Certain fundamental techniques presented by Sahni [21] and Ibarra and Kim [7] have been used by later researchers for designing $\varepsilon$-approximation algorithms for the CSP problem. To the best of our knowledge, Warburton [25] was the first to develop a fully polynomial time approximation algorithm for the CSP problem on acyclic networks. Hassin [5] later improved upon this to derive two fully polynomial time approximation schemes (FPAS). His methods are applicable for general networks. The first one is based on a combination of dynamic programming and scaling/rounding and has a complexity of $O(\log\log(U/L)[mn\,\varepsilon^{-1} + \log\log(U/L)])$, where $m$ and $n$ are, respectively, the number of nodes and links in the network, and $U$ and $L$ are, respectively, an upper bound and a lower bound on the optimal cost. In a more recent work Lorenz and Raz [13] improved upon this result by giving a strongly polynomial time approximation scheme of complexity $O(mn\,(\log\log n + \varepsilon^{-1}))$. This is also applicable to general networks. The second algorithm of Hassin is based on the interval partitioning technique developed by Sahni [21]. This is applicable only to acyclic networks. In [17], Philips proposed another strongly polynomial time approximation scheme applicable for general networks. In a subsequent work, Hong, Chung and Park [6] drew attention to certain flaws in the second algorithm of Hassin and the algorithm of Philip's. Other related approximation schemes providing certain improvements to Hassin's algorithm may be found in [12]. In another interesting paper [3], the authors considered the problem of determining a delay sensitive path whose delay is at most $(1 + \varepsilon)$ times the specified delay bound and whose cost is no greater than that of the minimum cost path of the CSP problem.

As regards heuristics, several of them have appeared in the literature providing different levels of performance with regard to the quality of the solution as well as the computation time required. For instance, the LHWHM algorithm [14] is a simple heuristic which is very fast (requiring only two invocations of Dijkstra's shortest path algorithm for a feasible problem). Reference [19] also discusses further enhancements of the LHWHM algorithm as well as a heuristic based on the Bellman-Ford-Moore (BFM) algorithm for the shortest path problem. It should be emphasized that in all these cases, only simulations are used to evaluate the performance of the algorithms. Usually, theoretical analysis is not given as regards the quality of the solution. A comprehensive overview of a number of quality of service routing algorithms may be found in [2].

There are heuristics that are based on sound theoretical foundations. These algorithms are based on solutions to the integer relaxation or the dual of the integer relaxation of the CSP problem. To the best of our knowledge, the first such algorithm was reported in [4] by Handler and Zang. This is based on the geometric approach (also called the hull approach [16], [29]). More recently, in an independent work, Jüttner etc. [8] developed the LARAC algorithm which solves the Lagrangian relaxation of the CSP problem (Here, the Lagrangian relaxation method is equivalent to the dual method). In contrast to the geometric method, they used an algebraic approach. They also presented several interesting results relating to the structure of the optimal solutions of the Lagrangian relaxation. In another independent work, Blokh and Gutin [1] defined a general class of combinatorial optimization problems (that are called the MCRT problems, namely, Minimum Cost Restricted Time Combinatorial Optimization problems) of which the CSP problem is a special case, and proposed an approximation algorithm to this problem. In a recent work, Xiao etc. [26] drew attention to the fact that the algorithms in [4] and [8] are equivalent. Mehlhorn and Ziegelmann [16] and Ziegelmann [29] have also observed this equivalence and have developed several insightful results. They arrived at these results using the hull (geometric) approach. In view of this equivalence, we shall refer to these algorithms as the LARAC algorithm. The work in [26] also establishes certain results using the algebraic approach. These results also hold true in the case of the general optimization problem considered in [1]. In another independent work, Xue [28] also arrived at the LARAC algorithm using the primal-dual method of linear programming. A more recent variant of these approaches may be found in [11]. As regards computational complexity, in [9], Jüttner proves the strong polynomiality of the LARAC algorithm, both for the general case and for the CSP problem. He has used certain results from the general area of fractional combinatorial optimization. An application of the parametric search method to the general class of combinatorial optimization problems involving two additive parameters may be found in [10]. Radzik [18] gives an excellent exposition of approaches to fractional combinatorial optimization problems. Binary search based algorithms for the integer relaxation of the CSP problem are discussed in [11], [26] and [29]. They also establish the polynomial complexity of this approach using geometric and algebraic methodologies, respectively. Several interesting algorithms related to the CSP problem and motivated by applications have appeared in the literature. For examples, see [12] and [20].

The organization of the rest of the paper is as follows. In Section II, we present the CSP problem and the general class of optimization problems, namely the MCRT problem [1], and point out the equivalence of the LARAC algorithm and the MCRT algorithm. In Section III we present an algebraic study of the integer relaxation of the CSP problem. In view of the equivalence of the

LARAC and the MCRT algorithms, one would expect the results in [8], though originally intended for the CSP problem, to hold true for the MCRT problem. In Section III, we establish these results and certain new results for the general case without involving the properties of shortest paths. These results provide the basis for other algorithms considered in later sections. In Section IV, we present a generalized version of an optimality condition presented in Section III. This condition is for the case of combinatorial optimization problems which involve more than one additive constraint. In Section V, we present a binary search approach for the CSP problem and also show that both the LARAC algorithm and this algorithm can be embedded with a tuning parameter whose value can be specified in advance depending on the allowable deviation of the cost of the path produced from the optimal cost. In Section VI, we develop a strongly polynomial time algorithm for the integer relaxation of the CSP problem. This is based on the parametric approach developed by Megiddo [15] for fractional combinatorial optimization problems. Finally in Section VII, we show how the LARAC algorithm can be integrated with $\varepsilon$-approximation techniques to achieve considerable speedup of approximation algorithms. Simulation results demonstrating the value of the integrated approach are also presented. We conclude in Section VIII summarizing our contributions. In addition to these contributions, the paper also provides a tutorial and a unified view of approaches for the integer relaxation of the CSP problem and its general version using an algebraic approach.

## II. The Constrained Shortest Path (CSP) Problem and Generality of the LARAC Algorithm

In this section, we first define the CSP problem and present the LARAC algorithm of [8]. We then define the general class of optimization problems (of which the CSP problem is a special case) considered in [1] and the MCRT algorithm also presented in [1]. We show the equivalence of the LARAC and the MCRT algorithms, thereby establishing the generality of the LARAC algorithm for solving combinatorial problems involving two metrics. We emphasize that the LARAC and the MCRT algorithms solve the integer relaxation of the CSP problem and not the CSP problem itself.

As pointed out by Mehlhorn and Ziegelmann [16], the LARAC algorithm can also be derived by the hull approach. In the course of the development of the LARAC algorithm, the authors of [8] established certain interesting claims without proofs. Using an algebraic approach (in contrast to the geometric ideas used in the hull approach), we establish that all these results hold in the general case too. We also present some other results which throw insight into the structure of the optimal solutions of the integer relaxation of the CSP problem.

**Constrained Shortest Path Problem (CSP):** Consider a network $G(N, E)$. Each link $(u, v) \in E$ is associated with two weights $c_{uv} > 0$ (say, cost) and $d_{uv} > 0$ (say, delay). Also are given two distinguished nodes $s$ and $t$ and $T > 0$. Let $P_{st}$ denote the set of all $s$-$t$ paths and for any path $p$, define

$$c(p) = \sum_{(u,v)\in p} c_{uv} \;\; and \;\; d(p) = \sum_{(u,v)\in p} d_{uv} \, .$$

Given $T > 0$, let $P_{st}(T)$ be the set of all the $s$-$t$ paths $p$ such that $d(p) \leq T$. A path in the set $P_{st}(T)$ is called a feasible path. The CSP problem is to find a path $p^* = \arg\min\{c(p) | \, p \in P_{st}(T)\}$. In other words, the CSP problem is to find a minimum cost feasible path. It can be formulated as the following integer linear program.

**CSP:**

Minimize $\sum_{(u,v)\in E} c_{uv} x_{uv}$

subject to $\forall\ u \in N$,

$$\sum_{\{v|(u,v)\in E\}} x_{uv} - \sum_{\{v|(v,u)\in E\}} x_{vu} = \begin{cases} 1 & for \quad u = s \\ -1 & for \quad u = t \\ 0 & otherwise \end{cases}$$

$$\sum_{(u,v)\in E} - d_{uv} \cdot x_{uv} - w = -T, w \geq 0$$

$$x_{uv} = 0 \ or \ 1, \forall (u,v) \in E$$

The CSP problem is known to be NP-hard [24]. The main difficulty lies with the integrality condition that requires that the variables $x_{uv}$ be 0 or 1. Removing or relaxing this requirement from the above integer linear program and letting $x_{uv} \geq 0$ leads to RELAX-CSP, the relaxed CSP problem. It is often convenient to solve the dual of the relaxed form of the CSP problem which we present below.

The dual involves $s$-$t$ paths and a variable $\lambda \geq 0$. For each link $(u, v)$, let the aggregated cost $c_\lambda$ be defined as $c_{uv} + \lambda\ d_{uv}$. For a given $\lambda$, let $c_\lambda(p)$ denote the aggregated cost of the path $p$. Finally define $L(\lambda)$ as:

$$L(\lambda) = \min\{c_\lambda(p) |\ p \in P_{st}\} - \lambda\ T. \tag{1}$$

Note that in the above, $\min\{c_\lambda(p) |\ p \in P_{st}\}$ is the same as the minimum aggregated cost of an $s$-$t$ path with respect to a given value of $\lambda$. This can be easily obtained by applying Dijkstra's algorithm using aggregated link costs. Let the $s$-$t$ path which has minimum aggregated cost with respect to a given λ be denoted as $p_\lambda$. Then $L(\lambda) = c_\lambda(p_\lambda) - \lambda T$ and the dual of the RELAX-CSP can be presented in the following form.

**DUAL-RELAX-CSP**: Find $L^* = \max \{L(\lambda) \mid \lambda \geq 0\}$.

We note that the problem of maximizing $L(\lambda)$ as above is also called the Lagrangian dual problem. The value of λ that achieves the maximum $L(\lambda)$ in DUAL-RELAX-CSP will be denoted by λ*. Note that $L^*$, the optimum value of DUAL-RELAX-CSP is a lower bound on the optimum cost of the path solving the corresponding CSP problem. The key issue in solving DUAL-RELAX-CSP is how to search for the optimal λ and determining the termination condition for the search. The LARAC algorithm of [8] presented in Fig. 1 is one such efficient search procedure.

Procedure LARAC($s$, $t$, $d$, $T$)
  $p_c := Dijkstra(s,t,c)$
  if $d(p_c) \leq T$ then return $p_c$
  $p_d := Dijkstra(s,t,d)$
  if $d(p_d) > T$ then return "there is no solution"
  repeat
    $\lambda := \dfrac{c(p_c) - c(p_d)}{d(p_d) - d(p_c)}$
    $r := Dijkstra(s,t,c_\lambda)$
    if $c_\lambda(r) = c_\lambda(p_c)$ then return $p_d$
    else if $d(r) \leq T$ then $p_d := r$ else $p_c := r$
  end repeat
end procedure

Fig. 1. LARAC algorithm

**Description of the algorithm:** In the LARAC algorithm of Fig. 1, *Dijkstra*($s$, $t$, $c$), *Dijkstra*($s$, $t$, $d$), and *Dijkstra*($s$, $t$, $c_\lambda$) denote, respectively, Dijkstra's shortest path algorithm using link costs, link delays, and combined link weights with respect to the multiplier λ.

1. In the first step, the algorithm calculates the shortest path on link costs. If the path found meets the delay constraint, this is surely the optimal path. Otherwise, the algorithm stores the path as the latest infeasible path, simply called the $p_c$ path. Then it determines the shortest path on link delays denoted as $p_d$. If $p_d$ is infeasible, there is no solution to this instance.
2. Set $\lambda = (c(p_c) - c(p_d))/(d(p_d) - d(p_c))$. With this value of $\lambda$, we can find a new $c_\lambda$-minimal path $r$. If $c_\lambda(r) = c_\lambda(p_c)$ ( $= c_\lambda(p_d)$), we have obtained the optimal $\lambda$ according to Claim 5 of [8]. Otherwise, set $r$ as the new $p_c$ or $p_d$ according to whether $r$ is infeasible or feasible.

**Minimum Cost Restricted Time Combinatorial Optimization (MCRT) Problem:** The MCRT problem as defined in [1] is as follows. Given a finite set $P$, finite family set $S$ of subsets of $P$, non-negative threshold $h$, and two non-negative real-valued functions $y$: $P \rightarrow R^+$ (say, cost) and $x$: $P \rightarrow R^+$ (say, delay). The MCRT problem is to seek a solution $F^* = \arg\min\{y(F) | F \in S, x(F) \leq h\}$, where $z(G) = \sum_{g \in G} z(g)$ for $z \in \{x, y\}$ and $G \in S$.

Evidently, the CSP problem is a special case of the MCRT problem and so the MCRT problem is also NP hard. Therefore, we consider solving the integer relaxation of the MCRT problem. This is achieved by the MCRT algorithm given in [1] and presented in Fig. 2. In this algorithm, it is assumed that there is an effective algorithm $A(a, b)$ for the corresponding minimum cost problem with respect to $a\,x(p) + b\,y(p)$, $p \in S$, where $a$, $b$ are the multipliers. For instance, in the case of the CSP problem, Dijkstra's algorithm for the minimum cost path problem can play the role of algorithm $A$. In Fig. 2, algorithm $A(a, b)$ returns $p = \arg\min\{ax(r) + by(r) | r \in S\}$.

```
Procedure MCRT (h)
  F := A(0, 1)
  if x(F) ≤ h then return F.
  H := A(1, 0)
  if x(H) > h then return "no solution"
  repeat
    a := y (H) – y(F)
    b := x(F) – x(H)
    c := x(F)y(H) – x(H)y(F)                        (a)
   G := A(a, b)
    if c = ax(G) + by(G) then                       (b)
        if x(G) ≤ h then return G  else return H
    if c > ax(G) + by(G) then                       (c)
        if x(G) ≤ h then H := G  else F := G.
  end repeat
end procedure
```

Fig. 2. MCRT algorithm

**Equivalence of LARAC and MCRT Algorithms:** Following the definition of the variables in Fig. 1 and Fig. 2, it can be seen that $H$ corresponds to $p_d$ while $F$ corresponds to $p_c$ and $\lambda$ corresponds to *a/b* because

$$\frac{a}{b} = \frac{y(H) - y(F)}{x(F) - x(H)}.$$

Furthermore, $\frac{c}{b} = \frac{x(F)y(H) - x(H)y(F)}{x(F) - X(H)} = \frac{y(H) - y(F)}{x(F) - x(H)} x(F) + y(F) = y(F) + \frac{a}{b} x(F).$

If the expressions (a), (b) and (c) in procedure MCRT are scaled by $b$, the MCRT algorithm reduces to the LARAC algorithm. In view of the equivalence of the LARAC algorithm and the MCRT algorithm, in the rest of the paper we shall refer to both these algorithms as simply LARAC.

To conclude this section, to the best of our knowledge, the LARAC algorithm was first presented in [4]. More recently, Xue [28] presented another variant of this algorithm. Mehlhorn and Ziegelmann [16] and Ziegelmann [29] point out that the algorithm as presented in [4] can be derived from what they call the hull approach. Blokh etc. [1] also use geometric ideas in developing the MCRT algorithm. On the other hand, Jüttner etc. [8] developed this algorithm using a purely algebraic approach.

## III. An Algebraic Study of the Relax-CSP Problem and its Generalization

The LARAC algorithm as developed in [8] was originally intended for the CSP problem. In view of its generality as discussed in the previous section, one would expect that the claims in [8] on which the LARAC algorithm is based do not depend on the properties of shortest paths. In other words, we would like to establish these claims without invoking properties of shortest paths. This is indeed true. In this section, we will present proofs of some of these claims for the sake of completeness. Furthermore, in the following section we also establish certain other new results that throw much insight into the structure of the solutions of the DUAL-RELAX-CSP problem. Though our proofs below do not involve shortest paths or their properties, we have decided to retain the terms such as "minimal path" whose interpretation in the general context should be obvious.

**Claim 1[8]**: Let $L(\lambda) = \min\{c_\lambda(p) | \; p \in P_{st}\} - \lambda\, T$. Then $L(\lambda)$ is a lower bound to the optimum objective of the CSP problem for any $\lambda \geq 0$.

**Claim 2[8]**: $L$ is a concave piecewise linear function, namely, the minimum of the linear functions $c(p) + \lambda(d(p) - T)$ for all $p \in P_{st}$.

**Claim 3[8]:** For any $\lambda \geq 0$ and $c_\lambda$-minimal path $p_\lambda$, $d(p_\lambda)$ is a supgradient of $L$ in the point $\lambda$.

**Claim 4**[8]: If $\lambda < \lambda^*$, then $d(p_\lambda) \geq T$ and if $\lambda > \lambda^*$, then $d(p_\lambda) \leq T$ for each $c_\lambda$-minimal path $p_\lambda$.

***Proof***: Let $p$ and $p^*$ denote a $c_\lambda$-minimal path and $c_{\lambda^*}$-minimal path respectively

$$L(\lambda^*) = c(p^*) + \lambda^* d(p^*) - \lambda^* T \leq c(p) + \lambda^* d(p) - \lambda^* T = L(\lambda) + (\lambda^* - \lambda)(d(p) - T).$$

Since $L(\lambda^*) \geq L(\lambda)$, $(\lambda^* - \lambda)(d(p) - T) \geq 0$.

Therefore, if $\lambda < \lambda^*$ then $d(p_\lambda) \geq T$ and if $\lambda > \lambda^*$ then $d(p_\lambda) \leq T$ for each $c_\lambda$-minimal path $p_\lambda$. ■

**Claim 5**[8]: A value $\lambda > 0$ maximizes the function $L(\lambda)$ if and only if there are paths $p_c$ and $p_d$ which are both $c_\lambda$-minimal and for which $d(p_c) \geq T$ and $d(p_d) \leq T$ ($p_c$ and $p_d$ can be the same, in this case $d(p_d) = d(p_c) = T$).

***Proof***: **a)** Proof of ***only if*** part: Suppose $\lambda$ is the optimal value that maximizes $L(\lambda)$. Let $p$ be the corresponding $c_\lambda$-minimal path and thus $L(\lambda) = c(p) + \lambda(d(p) - T)$. Without loss of generality, we only consider the case $d(p) > T$. If the $\lambda$ is slightly increased to $\lambda'$ ($> \lambda$), $c(p) + \lambda\,(d(p) - T)$ is also increased. Since $L(\lambda)$ is optimal, $p$ cannot be the $c_{\lambda'}$-minimal path any more; otherwise $L(\lambda') > L(\lambda)$. Let $p'$ be the new $c_{\lambda'}$-minimal path. If $|\lambda - \lambda'|$ is small enough, $p'$ is also the $c_\lambda$-minimal path because there are only a finite number of paths. It follows that $c(p') + \lambda'(d(p') - T) = L(\lambda') \leq L(\lambda) = c(p') + \lambda\,(d(p') - T)$.

Hence $\lambda'\,(d(p') - T) \leq \lambda\,(d(p') - T) \Rightarrow d(p') \leq T$ since $\lambda' > \lambda$.

Let $p_c = p$ and $p_d = p'$ completing the proof of the ***only if*** part.

**b)** Proof of ***if*** part**:** Let $p_c$ and $p_d$ be two $c_\lambda$-minimal paths and $d(p_c) \geq T$ and $d(p_d) \leq T$. Without loss of generality, assume $\lambda^*$ maximizes the function $L(\lambda^*)$ and $\lambda^* > \lambda$.

Since $\lambda < \lambda^*$, $d(p_c) \geq T$ and $d(p_d) \leq T$, it follows that $d(p_d) = T$.

Let $p^*$ denote the $c_{\lambda^*}$-minimal path. Then,

$$\begin{aligned} L(\lambda^*) &= c(p^*) + \lambda^* d(p^*) - \lambda^* T \leq c(p_d) + \lambda^* d(p_d) - \lambda^* T \\ &= L(\lambda) + (\lambda^* - \lambda)(d(p_d) - T) \leq L(\lambda) \end{aligned}$$

Therefore, $L(\lambda) = L(\lambda^*)$, which proves that $\lambda$ maximizes $L(\lambda)$. ■

**Claim 6[8]**: Let $0 \leq \lambda_1 < \lambda_2$, and $p_{\lambda_1}, p_{\lambda_2} \in P_{st}$ be $c_{\lambda_1}$-minimal and $c_{\lambda_2}$-minimal paths. Then $c(p_{\lambda_1}) \leq c(p_{\lambda_2})$ *and* $d(p_{\lambda_1}) \geq d(p_{\lambda_2})$.

***Proof***: Note that $c_\lambda(p) = c(p) + \lambda\, d(p)$.

Because $p_{\lambda_1}, p_{\lambda_2} \in P_{st}$ are $c_{\lambda_1}$-minimal and $c_{\lambda_2}$-minimal paths

$$c_{\lambda_1}(p_{\lambda_1}) \leq c_{\lambda_1}(p_{\lambda_2}) \Leftrightarrow c(p_{\lambda_1}) + \lambda_1 d(p_{\lambda_1}) \leq c(p_{\lambda_2}) + \lambda_1 d(p_{\lambda_2}) \text{, and}$$

$$c_{\lambda_2}(p_{\lambda_1}) \geq c_{\lambda_2}(p_{\lambda_2}) \Leftrightarrow c(p_{\lambda_1}) + \lambda_2 d(p_{\lambda_1}) \geq c(p_{\lambda_2}) + \lambda_2 d(p_{\lambda_2}) \text{. Then}$$

$$\begin{aligned} &(\lambda_1 - \lambda_2) d(p_{\lambda_1}) \leq (\lambda_1 - \lambda_2) d(p_{\lambda_2}) \Rightarrow d(p_{\lambda_1}) \geq d(p_{\lambda_2}) \\ &c(p_{\lambda_1}) \leq c(p_{\lambda_2}) + \lambda_1 [d(p_{\lambda_2}) - d(p_{\lambda_1})] \leq c(p_{\lambda_2}). \end{aligned}$$

Hence the claim holds. ■

The convergence of the LARAC algorithm is guaranteed by the following result.

**Claim 7** [8]**:** Let $p_c^1, p_c^2, p_c^3, \ldots$ and $p_d^1, p_d^2, p_d^3, \ldots$ denote the sequences of paths generated by the LARAC algorithm. Then

$$d(p_c^1) > d(p_c^2) > d(p_c^3) > \ldots > T \text{ and } d(p_d^1) < d(p_d^2) < d(p_d^3) < \ldots \le T .$$

***Proof***: Suppose $p_c$ and $p_d$ are the current paths in the LARAC algorithm with $\lambda_c$ and $\lambda_d$ as the corresponding $\lambda$ values. Suppose that neither of these two $\lambda$ values is the maximizing value.

Let $\lambda = \dfrac{c(p_c) - c(p_d)}{d(p_d) - d(p_c)}$ and $p_\lambda$ be the corresponding $c_\lambda$-minimal path.

Evidently, $c_\lambda(p_c) = c_\lambda(p_d)$ (recalling that $c_\lambda(p) = c(p) + \lambda\, d(p)$).

Suppose $\lambda$ is not the maximizing value either; otherwise, the algorithm stops immediately. We also have

$$c(p_c) + \lambda_c\, d(p_c) \le c(p_d) + \lambda_c\, d(p_d),$$

$$c(p_c) + \lambda_d\, d(p_c) \ge c(p_d) + \lambda_d\, d(p_d).$$

In fact, the equality cannot hold because neither $\lambda_c$ nor $\lambda_d$ is the maximizing multiplier.

So $\qquad \lambda_c < \dfrac{c(p_c) - c(p_d)}{d(p_d) - d(p_c)} = \lambda < \lambda_d .$

Consider 2 cases:

1) $d(p_\lambda) \le T$: In this case, because $d(p_\lambda) \ge d(p_d)$ by Claim 6, it suffices to show that $d(p_\lambda) \ne d(p_d)$.

Assume $d(p_\lambda) = d(p_d)$. Consider the following inequalities

$$c(p_\lambda) + \lambda\, d(p_\lambda) \le c(p_d) + \lambda\, d(p_d) \text{ and } c(p_\lambda) + \lambda_d\, d(p_\lambda) \ge c(p_d) + \lambda_d\, d(p_d).$$

Because $d(p_\lambda) = d(p_d)$, it follows that $c(p_\lambda) = c(p_d)$. Hence $c_\lambda(p_c) = c_\lambda(p_d) = c_\lambda(p)$, which implies that $\lambda$ is the maximizing value. This contradiction establishes the theorem.

2) $d(p_\lambda) > T$: Proof in this case follows along the same lines as above. ■

**Theorem 1**: Consider the problem:

$$\text{Minimize } y\, c(p_d) + (1-y)\, c(p_c) \tag{2}$$

$$\text{subjects to } \quad y\, d(p_d) + (1-y)\, d(p_c) = T \text{ and } 0 \le y \le 1, \tag{3}$$

where $p_c$ and $p_d$ are two $s$-$t$ paths such that $d(p_d) > T$ and $d(p_c) < T$.

Let $\lambda = \dfrac{c(p_d) - c(p_c)}{d(p_c) - d(p_d)}$ and suppose that for all $s$-$t$ path $p$, $d(p) \ne T$.

Then $p_d$ and $p_c$ minimize (2) if and only if they both are $c_\lambda$-minimal.

***Proof***: First, we prove that

$$y\, c(p_d) + (1-y)\, c(p_c) \ge L(\xi),\ \xi \in R^+. \tag{4}$$

In fact,

$$
\begin{aligned}
&L(\xi) = \min\{c_{\xi}(p) \mid p \in P(s,t)\} - \xi T \\
&\leq y\, c_{\xi}(p_d) + (1-y)\, c_{\xi}(p_c) - \xi\,(y\, d(p_d) + (1-y)\, d(p_c)) \\
&= y\,(c_{\xi}(p_d) - \xi\, d(p_d)) + (1-y)(c_{\xi}(p_c) - \xi\, d(p_c)) \\
&= y\, c(p_d) + (1-y)\, c(p_c).
\end{aligned}
$$

Using (3), (2) can be rewritten as:

$$y\, c(p_d) + (1 - y)\, c(p_c) = c(p_c) + \lambda\, (d(p_c) - T) = c(p_d) + \lambda\, (d(p_d) - T). \tag{5}$$

Evidently, $d(p_c) \neq T$ and $d(p_d) \neq T$.

a) Proof of the *if* part: Suppose $p_d$ and $p_c$ are $c_\lambda$-minimal paths. Then

$$L(\lambda) = c(p_c) + \lambda\, (d(p_c) - T) = y\, c(p_d) + (1 - y)\, c(p_c),$$

where $y\, d(p_d) + (1 - y)\, d(p_c) = T$, $0 \leq y \leq 1$. So (2) is minimized.

b) Proof of the *only if* part: Suppose $p_d$ and $p_c$ minimize (2) or rather (5). Assume $p$ is a $c_\lambda$-minimal path and $p_d$ and $p_c$ are not $c_\lambda$-minimal. Consider the case when $p$ is infeasible (If $p$ is feasible, the theorem can be proven similarly). We have

$$c(p) + \lambda d(p) < c(p_d) + \lambda d(p_d). \tag{6}$$

Then

$$\lambda' = \frac{c(p_d) - c(p)}{d(p) - d(p_d)} > \lambda\ .$$

Thus

$$
\begin{aligned}
&y'\, c(p_d) + (1 - y')\, c(p) = c(p_d) + \lambda'(d(p_d) - T) \\
&< c(p_d) + \lambda\,(d(p_d) - T) = y\, c(p_d) + (1 - y)\, c(p_c),
\end{aligned}
$$

where $y' d(p_d) + (1 - y') d(p) = y\, d(p_d) + (1 - y) d(p_c) = T$.

The contradiction above proves that $p_c$ and $p_d$ are $c_\lambda$-minimal paths. ■

From the above proof, it can be shown that the value of λ defined by the optimal solution $p_c$ and $p_d$ of (2) is equal to the maximizing λ searched by LARAC algorithm. Also the optimum value of RELAX-CSP is equal to the optimum value $L(\lambda^*)$ of DUAL-RELAX-CSP.

There may be more than one maximizing λ. Assume that there is some multiplier λ such that the delay of the corresponding path $p_\lambda$ is equal to the delay bound. In this case, an interval will serve as the maximizing multiplier and we can find the actual optimal path for the original CSP problem with that λ, recalling that $c(p_\lambda) = L(\lambda)$ which is the lower bound on the cost of the actual optimal path.

**Theorem 2**: If $\exists\ \lambda$ and the corresponding path $p_\lambda$ such that $d(p_\lambda) = T$, the maximizing $\lambda$ is one unique interval (maybe just one point); Otherwise, the maximizing $\lambda^*$ is unique.

***Proof***: This is a direct consequence of the concavity of the function $L(\lambda)$ stated in Claim 2. ■

**Theorem 3**: Given λ1 and λ2, such that $d(p_{\lambda 1}) > T \geq d(p_{\lambda 2})$. If we start the LARAC algorithm by initializing $p_c$ and $p_d$ as $p_{\lambda 1}$ and $p_{\lambda 2}$, respectively, then the LARAC algorithm finds a maximizing multiplier λ* satisfying λ1 < λ* ≤ λ2.

## IV. Characterization of Optimal Solutions of the Integer Relaxation of the General CSP($k$) Problem with $k > 1$ Additive Constraints

Consider a directed graph $G(N, E)$. Each link $(u, v)$ is associated with a set of $k + 1$ additive non-negative integer weights $\boldsymbol{C_{uv}} = (c_{uv}, w^1_{uv}, w^2_{uv}\ldots, w^k_{uv})$. Here $c_{uv}$ is called the cost of link $(u, v)$ and $w^i_{uv}$ is called the $i^{th}$ delay of link $(u, v)$. For path $p$ define

$$c(p) \equiv \sum_{(u,v)\in p} c_{uv} \text{ and } d_i(p) \equiv \sum_{(u,v)\in p} w^i_{uv},\ \ i = 1,..,k\,.$$

The value $c(p)$ is called the cost of path $p$ and $d_i(p)$ is called the $i$th delay of path $p$. Given $k$ positive integers $r_1, r_2\ldots, r_k$, an $s$-$t$ path is called feasible if $d_i(p) \leq r_i$ for $i = 1, 2\ldots k$ ($r_i$ is called the bound on the $i$th delay of a path). The CSP($k$) problem is to find a minimum cost feasible $s$-$t$ path.

Starting with an ILP formulation of the CSP($k$) problem and relaxing the integrality constraints we get the RELAX-CSP($k$) problem below. In this formulation, for each $s$-$t$ path $p$, we introduce a variable $x_p$.

**RELAX-CSP($k$)**

Minimize $\quad \sum_p c(p)x_p \quad$ (7)

subject to $\quad \sum_p x_p = 1 \quad$ (8)

$\sum_p d_i(p)x_p \leq r_i \ \ i = 1, \ldots, k \quad$ (9)

$x_p \geq 0, \forall\, p \in P_{st} \quad$ (10)

The dual of RELAX-CSP($k$) is given below.

**DUAL-RELAX-CSP($k$):**

$$\text{Maximize} \quad w - \lambda_1 r_1 \ldots - \lambda_k r_k \tag{11}$$

$$\text{subject to} \quad w - d_1(p)\,\lambda_1 \ldots - d_k(p)\,\lambda_k \le c(p)\,, \ \ \forall\, p \in P_{st} \tag{12}$$

$$\lambda_i \ge 0 \quad i = 1, \ldots, k \tag{13}$$

In the above dual problem $\lambda_1, \lambda_2 \ldots, \lambda_k$ and $w$ are the dual variables, with $w$ corresponding to (8) and each $\lambda_i$ corresponding to the $i$th constraint in (9).

It follows from (12) that $w \le c(p) + d_1(p)\,\lambda_1 \ldots + d_k(p)\,\lambda_k, \forall\, p \in P_{st}$. Since we want to maximize (11), the value of $w$ should be as large as possible, i.e.

$$w = \min_{p \in P_{st}} \{c(p) + d_1(p)\,\lambda_1 \ldots + d_k(p)\,\lambda_k\}.$$

With the vector $\boldsymbol{\Lambda}$ defined as $\boldsymbol{\Lambda} = (\lambda_1, \lambda_2 \ldots, \lambda_k)$, define

$$L(\boldsymbol{\Lambda}) = \min_{p \in P_{st}} \{c(p) + \lambda_1 (d_1(p) - r_1) \ldots + \lambda_k (d_k(p) - r_k)\}. \tag{14}$$

Notice that $L(\boldsymbol{\Lambda})$ is called the Lagrangian function in literature and is a concave continuous function of $\boldsymbol{\Lambda}$ .

Then DUAL-RELAX-CSP($k$) can be written as follows.

**DUAL-RELAX-CSP($k$): M**aximize $L(\boldsymbol{\Lambda})$, subject to $\boldsymbol{\Lambda} \ge 0$. (15)

The $\boldsymbol{\Lambda}^*$ that maximizes (15) is called the maximizing multiplier and is defined as

$$\boldsymbol{\Lambda}^* = \arg\max_{\Lambda \geq 0} L(\boldsymbol{\Lambda}). \tag{16}$$

We may use $L(\boldsymbol{\Lambda})$ as a lower bound of $c(p_{opt})$ to evaluate the quality of the approximate solution obtained by our algorithm. Here $p_{opt}$ refers to the optimum path for the CSP problem. Given $p \in P_{st}$ and $\boldsymbol{\Lambda}$, define

$$c_{\Lambda}(p) \equiv c(p) + d_1(p)\,\lambda_1 \ldots + d_k(p)\,\lambda_k,$$

$$d_{\Lambda}(p) \equiv d_1(p)\,\lambda_1 \ldots + d_k(p)\,\lambda_k\,.$$

Here $c_{\Lambda}(p)$ and $d_{\Lambda}(p)$ are called the aggregated cost and the aggregated delay of path $p$, respectively. We shall use $P_{\Lambda}$ to denote the set of $s$-$t$ paths attaining the minimum aggregated cost w.r.t. to $\boldsymbol{\Lambda}$. A path $p_{\Lambda} \in P_{\Lambda}$ is called a $\boldsymbol{\Lambda}$**-**minimal path.

**Theorem 4**: Given an instance of a feasible CSP($k$) problem, a vector $\boldsymbol{\Lambda} \geq 0$ maximizes $L(\boldsymbol{\Lambda})$ iff the following problem in the variables $u_j$ is feasible.

$$\sum_{p_j \in P_\Lambda} u_j d_i(p_j) = r_i,\ \forall i, \lambda_i > 0 \tag{17}$$

$$\sum_{p_j \in P_\Lambda} u_j d_i(p_j) \leq r_i,\ \forall i, \lambda_i = 0 \tag{18}$$

$$\sum_{p_j \in P_\Lambda} u_j = 1 \tag{19}$$

$$u_j \geq 0,\ \forall j,\ p_j \in P_\Lambda \tag{20}$$

***Proof***: **Sufficiency**: Let $\boldsymbol{x} = (u_1\ldots, u_r, 0, 0\ldots)$ be a vector of size $|P_{st}|$, where $r = |P_{\Lambda}|$. Obviously, $\boldsymbol{x}$ is a feasible solution to RELAX-CSP($k$). It suffices to show that $\boldsymbol{x}$ and $\boldsymbol{\Lambda}$ satisfy the complementary slackness conditions.

According to (12), $\forall p \in P_{st}$, $w \le c(p) + d_1(p)\,\lambda_1 \ldots + d_k(p)\,\lambda_k$. Since we need to maximize (11), the optimal $w = c(p) + d_1(p_\Lambda)\,\lambda_1 \ldots + d_k(p_\Lambda)\,\lambda_k\ \forall\ p_\Lambda \in P_\Lambda$. For all other paths $p$, $w - c(p) + d_1(p)\,\lambda_1 \ldots + d_k(p)\,\lambda_k < 0$. So $\boldsymbol{x}$ satisfies the complementary slackness conditions. By (17) and (18), $\boldsymbol{\Lambda}$ also satisfies complementary slackness conditions.

**Necessary**: Let $\boldsymbol{x}^*$ and $(w, \boldsymbol{\Lambda})$ be the optimal solution to RELAX-CSP($k$) and DUAL-RELAX-CSP($k$), respectively. It suffices to show that we can obtain a solution to (17)-(20) from $\boldsymbol{x}^*$.

We know that all the constraints in (12) corresponding to paths in $P_{st} - P_\Lambda$ are strict inequalities, and $w = c(p) + d_1(p_\Lambda)\,\lambda_1 \ldots + d_k(p_\Lambda)\,\lambda_k$, $\forall\ p_\Lambda \in P_\Lambda$. So, from complementary slackness conditions we get $x_p = 0$, $\forall\ p \in P_{st} - P_\Lambda$.

Now let us set $u_j$ corresponding to path $p$ in $P_\Lambda$ equal to $x_p$, and set all other $u_j$'s corresponding to paths not in $P_\Lambda$ equal to zero. The $u_i$'s so elected will satisfy (17) and (18) since these are complementary conditions satisfied by $(w, \boldsymbol{\Lambda})$. Since $x_i$'s satisfy (8), $u_j$'s satisfy (19). Thus we have identified a solution satisfying (17)-(20). ■

Let us consider the general problem with $k$ constraints denoted as MCRT($k$), which is the extension of the MCRT problem. The problem can be defined as: Given a finite set $P$, finite family set $S$ of subsets of $P$, non-negative threshold $h_i$, $i = 1, 2 \ldots k$, and $k + 1$ non-negative real-valued functions $y: P \rightarrow R^+$ (say, cost) and $x_i: P \rightarrow R^+$ (say, delays), $i = 1, 2 \ldots, k$. The MCRT($k$) problem is to seek a solution $F^* \in S$ with $y(F^*) = \min\{y(F) \mid F \in S, x_i(F) \le h_i, i = 1, 2 \ldots, k\}$, where $z(G) = \sum_{g \in G} z(g)$ for $z \in \{y, x_i, i = 1, 2 \ldots, k\}$ and $G \in S$.

We have proven an optimal condition for the RELAX-CSP($k$) problem (Theorem 4) and know that the CSP($k$) problem is a special case of MCRT($k$) problem. On the other hand, any MCRT($k$) problem can be transformed to a CSP($k$) problem. We can construct a network with source node $s$ and a sink $t$, and for any subset of $G \in S$, let a link be added from $s$ to $t$ with weight vector ($y(G)$, $x_1(G)$, $x_2(G)$…, $x_k(G)$) (see Fig. 3). Then the relaxation of the MCRT($k$) problem (without the integrality constraints) is equivalent to finding the minimal combined weight path from $s$ to $t$. Using this transformation we can obtain a generalized version of Theorem 4 applicable to the MCRT($k$) problem. However, notice that the transformation is valid only for establishing the optimality characterization.

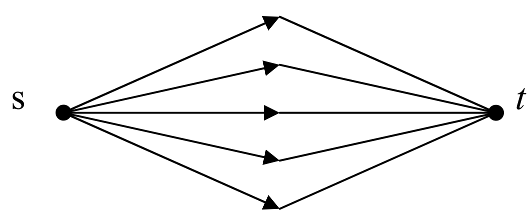



Fig. 3. Transformation of MCRT($k$) problem to CSP($k$) problem

## V. LARAC-BIN: A Binary Search Based Approach to the DUAL-RELAX-CSP Problem

In this section we present a new algorithm called LARAC-BIN that uses the binary search technique to find the maximizing multiplier. LARAC-BIN as presented in Fig. 4 stops when $L(\lambda^*) - L(\lambda) < \tau$. The parameter $\tau$ serves as a tuning parameter and can be specified in advance depending on the allowable deviation of the cost of the produced solution from the optimum value. We also establish an optimality condition. This criterion can be used to terminate the algorithm and at termination the optimum value of $L(\lambda)$ will be obtained.

**Procedure LARAC-BIN** $(s, t, T, \tau)$
$p_c := Dijkstra(s, t, c)$
**if** $d(p_c) \leq T$ **then return** $p_c$
$p_d := Dijkstra(s, t, d)$
**if** $d(p_d) > T$ **then return** "there is no solution"
**if** $d(p_d) = T$ **or** $c(p_d) = c(p_c)$ **then return** $p_d$
$\lambda_{begin} := 0, \lambda_{end} := (c(p_d) - c(p_c))/(T - d(p_d))$
**while** $(\lambda_{end} - \lambda_{begin})(T - d(p_d)) > \tau$
  $\lambda := (\lambda_{begin} + \lambda_{end})/2$
  $r := Dijkstra(s, t, c_\lambda)$
  **if** $d(r) = T$ **then return** $r$
  **else if** $d(r) < T$ **then** $\lambda_{end} := \lambda$ **else** $\lambda_{begin} := \lambda$
**end while**
**return** $r := Dijkstra(s, t, c_{\lambda_{end}})$
**end procedure**

Fig. 4. LARAC-BIN algorithm

In effect, the goal of the LARAC-BIN is to find the minimum λ with which we can obtain a feasible path because the smaller the λ, the smaller the cost of the path obtained. This goal is compatible with that of the LARAC algorithm searching for the maximizing λ* and $L(\lambda^*)$. To put it formally, we have the following theorem.

**Theorem 5**: Let λ* denote the smallest maximizing value for $L(\lambda)$ and $p_\lambda$ denote a path corresponding to λ. Then $c(p_{\lambda^*}) \leq c(p_\lambda)$ for all λ such that $d(p_\lambda) \leq T$.

***Proof***: According to Claim 6, if $\lambda^* \leq \lambda$, $c(p_{\lambda^*}) \leq c(p_\lambda)$. So assume $\lambda^* > \lambda$.

In this case, $d(p_\lambda) \leq T$ implies $d(p_\lambda) = T$ by Claim 4. Hence $L(\lambda) = L(\lambda^*)$ according to Claim 5, which is impossible because λ* is the smallest maximizing value for $L(\lambda)$.

The above contradiction proves the theorem. ■

The initial values of $\lambda_{begin}$ and $\lambda_{end}$ in Fig. 4 are to be selected such that $p_{begin}$ is infeasible and $p_{end}$ is feasible. We can initialize $\lambda_{end}$ as in the following theorem.

**Theorem 6**: If $\lambda = \dfrac{c(p_d) - c(p_c)}{T - d(p_d)}$, $d(p_d) < T$ and $c(p_d) > c(p_c)$, then the $c_\lambda$-minimal path is feasible, where $p_c$ and $p_d$ are the minimal cost and minimal delay path, respectively.

***Proof***: Assume that $p$ is a $c_\lambda$-minimal path and $d(p) > T$. It follows that

$$c(p_d) + \lambda\, d(p_d) \geq c(p) + \lambda\, d(p).$$

Then

$$0 \leq c(p_d) - c(p) - \frac{c(p_d) - c(p_c)}{T - d(p_d)}(d(p) - d(p_d)) < c(p_d) - c(p) - (c(p_d) - c(p_c)) = c(p_c) - c(p) \leq 0.$$

The above contradiction proves the theorem. ■

**Theorem 7**: Let $\lambda^*$ denote the smallest maximizing Lagrangian multiplier of $L(\lambda)$ and $p^*$ be the resulting path. Let $p_{begin}$ and $p_{end}$ be the minimal aggregated (combined) cost paths with respect to $\lambda_{begin}$ and $\lambda_{end}$, where $\lambda_{begin}$ and $\lambda_{end}$ are as defined in the LARAC-BIN algorithm in Fig. 4. Here $p_{begin}$ is infeasible and $p_{end}$ is feasible. Then $0 \leq L(\lambda^*) - L(\lambda_{end}) \leq (\lambda_{end} - \lambda_{begin})(T - d(p_{end}))$.

***Proof***: The left inequality holds because $L(\lambda^*)$ is the maximum value.

Evidently, $d(p_{end}) \le T$, $\lambda_{begin} \le \lambda^* \le \lambda_{end}$, and $c(p^*) + \lambda^* d(p^*) \le c(p_{end}) + \lambda^* d(p_{end})$.

It follows that

$$\begin{aligned}
&L(\lambda^*) - L(\lambda_{end}) = c(p^*) + \lambda^* d(p^*) - \lambda^* T - [c(p_{end}) + \lambda_{end} d(p_{end}) - \lambda_{end} T] \\
&= \{c(p^*) + \lambda^* d(p^*) - [c(p_{end}) + \lambda^* d(p_{end})]\} - (\lambda_{end} - \lambda^*) d(p_{end}) + (\lambda_{end} - \lambda^*) T \\
&\le (\lambda_{end} - \lambda^*)(T - d(p_{end})) \le (\lambda_{end} - \lambda_{begin})(T - d(p_{end})).
\end{aligned} \qquad \blacksquare$$

Note that we have used the result of the above theorem in the termination of the LARAC-BIN algorithm (Fig. 4).

Since a number of optimization problems only involve integer values (integer problem) or can be converted to integer problems, we now derive a termination condition for the LARAC-BIN algorithm when all the link costs and delays are integers. If terminated according to this condition, the algorithm computes the maximizing λ* with polynomial time complexity.

Consider the set of rational numbers $Q(D) = \{p / q \mid \mathrm{GCD}(p, q) = 1, q \le D, \text{ and } p, q, D \in N^+\}$. Define the density of $Q(D)$ as $\mathrm{DENS}(Q(D)) = \min\{|x_1 - x_2|: x_1, x_2 \in Q(D) \text{ and } x_1 \ne x_2\}$. It is easy to show that $\mathrm{DENS}(Q(D)) = 1/D^2$ and that for $x, y \in Q(D)$, $x = y$ if $|x - y| < \mathrm{DENS}(Q(D))$.

Suppose that we modify LARAC-BIN so that it terminates when $|\lambda_{begin} - \lambda_{end}| < 1 / D^2$ and that the paths at termination are $p_{end}$ and $p_{begin}$, where $D = |d(p_{begin}) - d(p_{end})|$. Let

$$\lambda = \lambda' = \frac{c(p_{end}) - c(p_{begin})}{d(p_{begin}) - d(p_{end})}.$$

**Theorem 8**: λ' defined as above is a maximizing multiplier.

***Proof***: Consider $Q(D)$, where $D = |d(p_{begin}) - d(p_{end})|$. Because

$$c(p_{begin}) + \lambda_{begin} d(p_{bebin}) \leq c(p_{end}) + \lambda_{begin} d(p_{end}) \text{ and}$$
$$c(p_{begin}) + \lambda_{end} d(p_{bebin}) \geq c(p_{end}) + \lambda_{end} d(p_{end}),$$

$$\lambda_{begin} \leq \lambda' = \frac{c(p_{end}) - c(p_{begin})}{d(p_{begin}) - d(p_{end})} \leq \lambda_{end}.$$

Suppose that $\lambda_{\text{begin}} \leq \lambda^* \leq \lambda_{end}$, where $\lambda^*$ is the maximizing Lagrangian multiplier obtained by LARAC algorithm initialized with $p_c = p_{begin}$ and $p_d = p_{end}$.

Clearly $\lambda^* = (c(p_{\lambda 1}) - c(p_{\lambda 2})) / (d(p_{\lambda 2}) - d(p_{\lambda 1}))$ for some paths $p_{\lambda 1}$ and $p_{\lambda 2}$ w.r.t. the Lagrangian multipliers $\lambda 1$ and $\lambda 2$. It can be seen that $\lambda 1$ and $\lambda 2 \in [\lambda_{begin}, \lambda_{end}]$ following the similar argument above. Hence $|\, d(p_{\lambda 2}) - d(p_{\lambda 1})| \leq D$ according to Claim 6, i.e., $\lambda^* \in Q(D)$.

Evidently $|d(p_{begin}) - d(p_{end})| = D \leq D$ and thus $\lambda \in Q(D)$.

Because $|\, \lambda' - \lambda^*| < |\lambda_{begin} - \lambda_{end}| < 1 / D^2 = \text{DENS}\ (Q(D))$, the only possibility is that $\lambda' = \lambda^*$. ■

For the CSP problem, the size of $D$ is bounded as $D \leq |\boldsymbol{N}| \max \{d_{ij} \mid (i, j) \in \boldsymbol{E}\}$. If the LARAC-BIN algorithm is terminated using the condition given above, then we have the following complexity result.

**Theorem 9**: LARAC-BIN terminates in $O((m + n \log n) (\log (COST \times DELAY^2)))$ time where $COST$ is the cost of the minimum delay path and $DELAY$ is the delay of the minimum cost path in the network.

## VI. Strong Polynomiality of DUAL-RELAX-CSP: A Parametric Search Based Algorithm

Jüttner [9] has shown that the LARAC algorithm for DUAL-RELAX-CSP is strongly polynomial. We wish to note that the time complexity of an algorithm for a graph/network problem is strongly polynomial if the computational time is a function of only $m$ and $n$, where $m$ and $n$ are respectively the number of links and the number of nodes in the graph/network. In this section, we present another strongly polynomial time algorithm, namely the PSCSP (Parametric Search Based Constrained Shortest Path) algorithm (Fig. 5), for solving DUAL-RELAX-CSP. This method is based on a methodology first proposed by Megiddo [15] to solve fractional combinatorial optimization problems. In this section, we only handle the shortest path problem without generalization due to the nature of the parametric search. The algorithm PSCSP in Fig. 5 is based on the BFM algorithm.

Let $\lambda^* \geq 0$ denote the maximizing Lagrangian multiplier for the $L(\lambda)$ function. Assume node 1 is the source node and node $n$ is the sink node. Each node $v$ of the network is associated with a pair $M_v = (x_v, y_v)$, where $x_v$ and $y_v$ keep track of the cost and delay of some 1-$v$ path during the execution of the PSCSP algorithm. $M$ is initialized as $M_1 = (0, 0)$ and $Mv = (\infty, \infty)$ for $v \neq 1$. The algorithm computes the $c_{\lambda^*}$-minimal $1 - n$ path. This algorithm does not guarantee the feasibility of the obtained path. In order to get a feasible $c_{\lambda^*}$-minimal $1 - n$ path, we can revise the BFM algorithm using lexicographic ordering on the combined link costs and link delays [11, 22]. For the details of the algorithm computing a feasible $c_{\lambda^*}$-minimal $1 - n$ path, please refer to [27].

$$
\begin{array}{l}
\text{Procedure PSCSP}(s,t,T) \\
\quad M_v = (x_v, y_v) = (\infty,\infty) \text{ for } v = 2,\ldots n \\
\quad M_1 = (0,0) \\
\quad \text{for i} \leftarrow 1 \text{ to } n\text{ - 1 do} \\
\quad\; \text{for } u \leftarrow 1 \text{ to } n \text{ do} \\
\quad\quad \text{for each } v \text{ such that } (u,v) \in E \text{ do} \\
\quad\quad\; *[\;\text{if } (x_v + \lambda * \mathrm{y}_v \\
\quad\quad\quad\quad\quad \geq x_u + \lambda * y_u + c_{uv} + \lambda * d_{uv})\,] \qquad\qquad (21) \\
\quad\quad\quad\quad M_v \leftarrow (x_u + c_{uv}, y_u + d_{uv})
\end{array}
$$

Fig. 5. PSCSP Algorithm ($\lambda^*$ is unknown)

In the algorithm in Fig. 5, we need extra steps to decide whether the Boolean expression (21) (it is called oracle test) is true or false since λ* is unknown.

If $x_v = \infty$, $y_v = \infty$, then the inequality holds. Assume $x_v$ and $y_v$ are finite non-negative values. Then it suffices to evaluate the following Boolean expression.

$$(x_u + c_{uv} - x_v) + \lambda^*(y_u + d_{uv} - y_v) = p + q\,\lambda^* \leq 0, \tag{22}$$

where $p = x_u + c_{uv} - x_v$ and $q = (y_u + d_{uv} - y_v)$.

If $p \cdot q \geq 0$, then it is trivial to tell whether (22) holds or not. Suppose $p \cdot q < 0$, i.e., $-p/q > 0$.

Let $\lambda = -p/q$ and let $r = Dijkstra(s, t, c_\lambda)$, where *Dijkstra* computes a $c_\lambda$-minimal path.

Now consider three cases:

a) $d(r) > T$: By Claim 4 of Section III, $\lambda \leq \lambda^*$ and thus (22) can be decided according to whether $q$ is positive or negative.

b) $d(r) < T$: By Claim 4, $\lambda \geq \lambda^*$ and (22) can be evaluated similarly.

c) $d(r) = T$: Return the path $r$ as the optimal path (by Claim 5).

If PSCSP is based on Dijkstra's algorithm, instead of the BFM algorithm, the complexity of the resultant algorithm is reduced to $O((m + n \log n)^2)$. Thus we have the following result.

**Theorem 10**: The parametric search algorithm PSCSP for DUAL-RELAX-CSP is strongly polynomial with time complexity $O((m + n \log n)^2)$. ■

In the implementation of the PSCSP algorithm, the number of invocations of Dijkstra's algorithm is reduced by maintaining an interval $[a, b]$ containing $\lambda^*$, where $a$ is the maximum known value of $-p/q < \lambda^*$ and $b$ is the minimum known value of $-p/q > \lambda^*$ during the execution of the algorithm. We only need to call *Dijkstra* algorithm for $\lambda$ within the interval $[a, b]$ and update the interval accordingly. A discussion of the application of the parametric approach to the general class of optimization problems involving two additive parameters may be found in [10].

## VII. Closing the gap: An Integrated Approach to ε-Approximation Algorithm Design for the CSP Problem

In this section, we show how the LARAC algorithm can be used to considerably speed up an $\varepsilon$-approximation scheme. A few definitions are now in order.

An approximation algorithm for a minimization problem obtains a solution whose cost is within a specified multiple of the optimum cost. This idea is formally stated as follows [21].

An approximation scheme for a problem $P$ is an algorithm that, given an instance $I$ and a desired degree of accuracy $\varepsilon > 0$, constructs a problem solution with value $\hat{F}(I)$, such that, if $F^*(I) > 0$ is the value of an optimal solution to $I$, then

$$\frac{|F^*(I) - \hat{F}(I)|}{F^*(I)} \leq \varepsilon$$

A fully polynomial time approximation scheme for a graph/network optimization problem is an approximation scheme whose computing time is a polynomial function of the input size and $1/\varepsilon$. A strongly polynomial time approximation scheme for a graph/network optimization problem is an approximation scheme whose computing time is a polynomial function of the number of nodes and $1/\varepsilon$.

In the literature, there has been an extensive discussion of approximation algorithms for the CSP problem. Of particular interest to us are Hassin's algorithm [5] and the more recent algorithm due to Lorenz and Raz [13]. Hassin presents a fully polynomial time $\varepsilon$-approximation and Lorenz and Raz present a strongly polynomial time approximation scheme (SEA algorithm).

There are two phases in the design of approximation algorithms:

Phase1:

Start with an interval [$LB$, $UB$] where $LB$ and $UB$ are lower and upper bounds to the objective value of the optimum solution to the CSP problem, and iteratively shrink the interval until the ratio of the upper bound and the lower bound is below some constant (say, 2). This is achieved using a combination of a dynamic programming algorithm and a test procedure to determine whether the optimum is greater than or equal to a specified value.

Phase 2:

Determine an $\varepsilon$-approximate solution using the dynamic programming algorithm with the lower and upper bounds obtained in the phase 1.

Since LARAC/LARAC-BIN is very fast, we can use them to construct Phase 1. This considerably improves the computational time over the original $\varepsilon$-approximation algorithm which does not use LARAC for the first phase. The details of this integration are given below.

LARAC algorithm terminates with two paths $p_c$ and $p_d$ one of which is feasible, denoted by $p_d$, and the other is infeasible, denoted by $p_c$. It is easy to see that the cost of the infeasible path is the lower bound and the cost of the feasible path is the upper bound on the optimal cost. The value of $p_c$ at termination of LARAC is also a lower bound on the cost of the optimal path to the CSP problem. Given a parameter $\varepsilon$, if the cost of $p_d$ at termination is less than $(1 + \varepsilon)\ c(p_c)$, then $p_d$ is an $\varepsilon$-approximation to the CSP problem. If this is not the case, then the paths $p_c$ and $p_d$ can be used to get the initial lower and upper bounds required by $\varepsilon$-approximation algorithms. The integrated algorithm incorporating the above ideas is presented in Fig. 6. Here we have used the SEA algorithm presented in [13] for Phase 2.

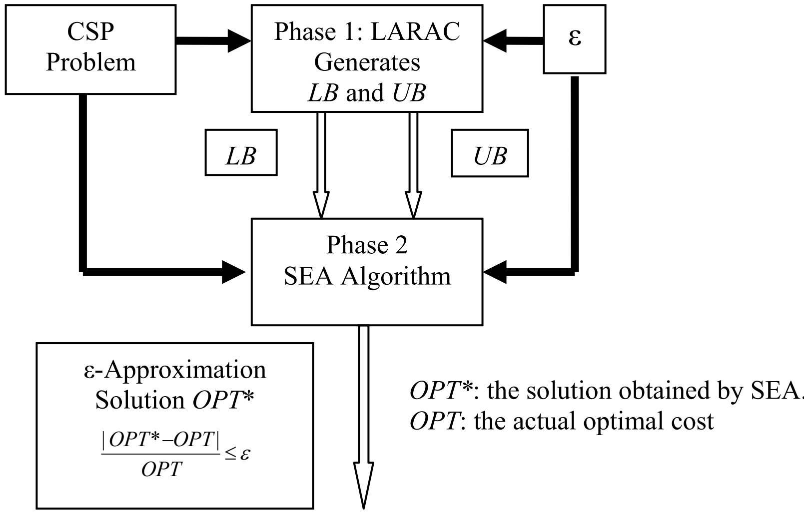
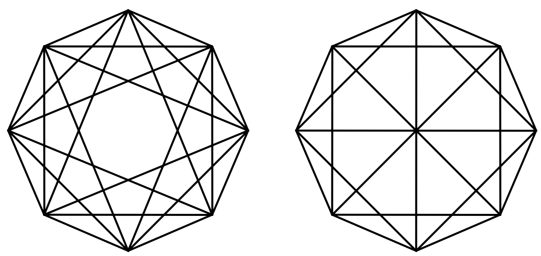



Fig. 6. An integrated approximation algorithm: LARAC + SEA



(a) $H_{6,8}$ (b) $H_{7,9}$

Fig. 7. $H_{k,n}$ graphs

We next discuss results of our simulation of the integrated approach. In our experiments we have used regular graphs $H_{k,n}$ (See Fig. 7) proposed by Harary (See [23]), where $k$ is the degree and $n$ is the number of nodes, respectively. The link costs are randomly generated integers in the range 2 to 198 and delays are assigned values as follows: $d_{ij} = 200 - c_{ij}$, where $c_{ij}$ and $d_{ij}$ are the cost and delay of link $(i, j)$, respectively. For each pair of vertex and degree, 10 experiments are carried out and the average value is given in Table 1.

As we can see from column six in the table, the ratio of the cost of $p_d$ and the cost of $p_c$ returned by LARAC is very close to 1. This is much better than the ratio of 2 which Phase 1 tries to achieve. Column seven shows that the total time for Phase 1 (when LARAC is used) is only about 5% of the total running time. We also note that Phase 1 when LARAC is used takes only 0.1% of the time for Phase 1 when the dynamic programming approach is used. Furthermore, we can also see from the last column in the table that the integrated approach achieves a speedup of 6.

**Table 1** Simulation Results

R = the ratio of the cost of $p_d$ and the cost of $p_c$ returned by LARAC
LT = the ratio of the time used by LARAC and the total running time (LARAC + SEA)
T = the ratio of the time used by LARAC+SEA and the time used by pure SEA algorithm

| NODE | DEG | ε | SEA | LARAC+SEA | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | Cost | LARAC | | | Cost | T |
| | | | | Cost | R | LT | | |
| 1000 | 6 | .05 | 13290 | 13290 | 1.1 | .005 | 13388 | .16 |
| 1000 | 16 | .05 | 9696 | 9748 | 1.1 | .002 | 9696 | .27 |
| 1000 | 32 | .05 | 5946 | 6196 | 1.2 | .004 | 5966 | .14 |
| 2000 | 6 | .05 | 28002 | 29888 | 1.1 | .002 | 28000 | .14 |
| 2000 | 16 | .05 | 19704 | 20222 | 1.1 | .003 | 19704 | .10 |
| 2000 | 32 | .05 | 11634 | 11778 | 1.1 | .002 | 11636 | .12 |

## VIII. Summary

In this paper, we have studied several aspects of the constrained shortest path (CSP) problem. This is an NP-complete problem and so in the literature, the focus has been on solving the integer relaxation of the problem called RELAX-CSP. We first pointed out the equivalence of the algorithms presented in [1], [4] and [8]. In view of this equivalence, we call these algorithms simply as the LARAC algorithm. Whereas the algorithms in [4] and [8] were intended for the CSP problem, the one in [1] was intended for a general class of combinatorial optimization problems (MCRT problem) involving two additive parameters. Using an algebraic approach, we have shown in Section III that all the claims in [8] also hold for the MCRT problem. We have also established certain new results on the properties of the solutions obtained by the LARAC algorithm. In particular, we have shown that the paths $p_c$ and $p_d$ that result at the termination of LARAC have an interesting property and, in fact, solve another optimization problem (Theorem 1). In Section IV, we generalize Claim 5 of Section III and develop a characterization of optimal solutions for the general CSP problem involving more than one additive constraint. This is also true for the generalized version of the MCRT problem involving more than one additive constraint.

In Section V, we presented a heuristic called LARAC-BIN based on binary search. The new heuristic involves a tuning parameter whose value can be specified in advance depending on the allowable deviation of the cost of the path produced by the heuristic from the optimum value. Whereas binary search is a commonly employed technique for algorithm design, incorporation of the tuning parameter as in LARAC-BIN enhances the value of the binary search based approaches.

In Section VI, we presented a strongly polynomial time algorithm for DUAL-RELAX-CSP. This algorithm is based on Megiddo's parametric search method [15] and certain techniques from fraction combinatorial optimization [18]. To the best of our knowledge, this algorithm has the best time complexity to date for DUAL-RELAX-CSP.

In Section VII, we pointed out how LARAC and LARAC-BIN can be used in conjunction with ε-approximation techniques to generate paths whose costs are guaranteed to be within certain factor of the optimum. The value of $L(\lambda)$ at termination of these algorithms is a lower bound on the cost of the optimum solution to the CSP problem. Given a parameter $\varepsilon$, if the cost of the path $p_d$ at termination is less than $(1 + \varepsilon)\ c(p_c)$, then $p_d$ is an $\varepsilon$-approximation to the CSP problem. If this is not the case, then the paths $p_c$ and $p_d$ can be used to generate lower and upper bounds needed for an $\varepsilon$-approximation algorithm. An integrated approach to the design of $\varepsilon$-approximation algorithms based on these ideas has been presented in Section VII. Effectiveness of this integrated approach has been illustrated through simulation.

Besides establishing new results, the paper also provides a tutorial on and a unified view of approaches for the CSP problem and its general version using an algebraic approach.

## References


[1] David Blokh and Georgia Gutin, "An Approximation Algorithm for Combinatorial Optimization Problems with Two Parameters," *Australasian Journal of Combinatorics*, vol. 14, 1996, pp.157-164.

[2] Shigang Chen and Klara Nahrstedt, "An Overview of Quality-of-Service Routing for the Next Generation High-speed Networks: Problems and Solutions," *IEEE Network Magazine*, 12(6), Nov. / Dec, 1998.

[3] Ashish Goel, K.G. Ramakrishnan, Deepak Kataria and Dimitris Logothetis, "Efficient Computation of Delay-sensitive Routes from One Source to All Destinations," *IEEE INFOCOM*-2001, pp 854-858.

[4] G.Handler and I. Zang, "A dual algorithm for the constrained shortest path problem," Networks 10, 293-310, 1980.

[5] R.Hassin, "Approximation Schemes for the Restricted Shortest Path Problem," *Mathematics of Operation Research*, 17(1), 1992, pp.36-42.

[6] Sung-Pi Hong, Sung-Jin Chung and Bum Hwan Park, "On 'Strongly' and Fully Approximation Schemes for Restricted Shortest Path Problem," Technical Report, Seoul National University(csj@optima.sun.ac.kr).

[7] O. Ibarra and C. Kim, "Fast Approximation Algorithms for the Knapsack and Sum of Subsets Problems," *Journal of the Association for Computing Machinery*, vol. 22, 463-468, October, 1975.

[8] Alpár Jüttner, Balázs Szviatovszki, Ildikó Mécs and Zsolt Rajkó, "Lagrange Relaxation Based Method for the QoS Routing Problem," in *IEEE INFOCOM*-2001, pp. 859-868.

[9] Alpár Jüttner, "On Resource Constrained Optimization Problems," *4th Japanese-Hungarian Symposium on Discrete Mathematics and Its Applications*, June 3-6, 2005, Budapest, Hungary.

[10] Alpár Jüttner, "On Budgeted Optimization Problems," under review for SIAM Journal on Discrete Mathematics.

[11] Turgay Korkmaz, Marwan Krunz and Spyros Tragoudas, "An Efficient Algorithm for Finding a Path Subject to Two Additive Constraints," *Computer Communications Journal* 25(3):225-238, 2002.

[12] Dean H. Lorenz, Ariel Orda, Danny Raz and Yuvait Shavitt, "Efficient QoS Partition and Routing of Unicast and Mulicast," in *IWQOS*, June 2000.

[13] D.Lorenz and D.Raz, "A Simple Efficient Approximation Scheme for the Restricted Shortest Paths Problem," *Operations Research Letters*, vol. 28, June 2001, pp. 213-219.

[14] Gang Luo, Kaiyuan Huang, Jianli Wang, Chris Hobbs and Ernst Munter, "Multi-QoS Constraints Based Routing for IP and ATM Networks," in *IEEE Workshop on QoS Support for Real-Time Internet Applications,* June 1, 1999, Vancouver Canada.

[15] N. Megiddo, "Combinatorial optimization with rational objective functions," *Math. Oper. Res.*, 4:414-424, 1979.

[16] K. Mehlhorn and M. Ziegelmann, "Resource Constrained Shortest Path," in the 8*th European Symposium on Algorithms* (*ESA* 2000).

[17] Cynthia Phillips, "The Network Inhibition Problem," in *the* 25*th* *Annual ACM Symposium on the Theory of Computing*, May, 1993.

[18] T. Radzik, 'Fractional Combinatorial Optimization', in Handbook of Combinatorial Optimization, Editors DingZhu Du and Panos Pardalos, vol. 1, Kluwer Academic Publishes, Dec. 1998.

[19] Ravi Ravindran, K.Thulasiraman, Anindya Das, Kaiyuan Huang, Gang Luo and Guoliang Xue, "Quality of Services Routing: Heuristics and Approximation Schemes with a Comparative Evaluation," in *ISCAS*, 2002.

[20] Douglas S. Reeves and Hussein F.Salama, "A Distributed Algorithm for Delay-Constrained Unicast Routing," *IEEE/ACM Trans. on Networking*, vol. 8, no 2, April 2000, pp. 239-250.

[21] Sartaj Sahni, "General Techniques for Combinatorial Approximation," *Oper. Res.* 25, 920-936, 1977.

[22] João Luís Sobrinho, "Algebra and Algorithms for QoS Path Computation and Hop-by-Hop Routing in the Internet," *IEEE/ACM Trans. on Networking*, vol. 10, no. 4, Aug. 2002.

[23] K. Thulasiraman and M. N. Swamy, *Graphs*: *Theory and Algorithms*, New York: Wiley Interscience, 1992.

[24] Z. Wang and J. Crowcroft, "Quality-of-Service Routing for Supporting Multimedia Applications," *IEEE on Selected Areas in Communications*, vol.14, no.7, pp. 1228-1234, September 1996.

[25] A. Warburton, "Approximation of pareto optima in multiple-objective shortest path problems," *Operations Research*, 35:70-79,1987.

[26] Y. Xiao, K. Thulasiraman and G. Xue, "Equivalence, Unification and Generality of Two Approaches to the constrained Shortest Path Problem with Extension," in *Allerton conference on Control, Communication and Computing*, *University of Illinois, Urbana-Champaign,* Oct. 1- 3, 2003, Urbana-Champaign, IL.

[27] Y. Xiao, K. Thulasiraman and G. Xue, "GEN-LARAC: A Generalized Approach to the Constrained Shortest Path Problem under Multiple Additive Constraints", in *International Symposium on Algorithms and Computation* (ISAAC), 2005.

[28] Guoliang Xue, "Minimum-Cost QoS Multicast and Unicast Routing in Communication Networks," *IEEE Trans. on Communications*, vol. 51, no.5, May 2003, pp.817-824.

[29] M. Ziegelmann, Constrained Shortest Paths and Related Problems, PhD thesis, Max-Planck Institute für Informatik, Germany, 2001 (http://www.mpi-sb.mpg.de/~mark/).