Combined Bus and Driver Scheduling

C. Valouxis, E. Housos

Computers and Operation Research Journal

Vol 29/3, pp 243-259, March 2002

Problem Definition (1)

- **<u>Shift</u>**: a set of routes that will be performed by a bus and its driver in one day
- Shifts must be legal according to a complex set of rules
- The solution of the problem is a set of shifts that cover all the work with the minimum cost

Problem Definition (2)

- Bus
  - Type
  - Kilometers
- Driver
  - Base, current position
  - Starting hour of work
  - Available days before the scheduled days-off

Problem Definition (3)

- Trip segment
  - Departure and Arrival time and place
  - Required bus type (fleet requirement)
  - Distance in km
- Shift
  - Set of trip segments and rest-time assigned to a specific bus and driver

Problem Definition (4)

- The solution of the problem is a set of shifts that must satisfy the following
  - All shifts should be legal
  - All required trip segments must be covered
  - Minimum cost
  - Attempt to balance monthly total overtime & kilometer parameters

Integer Program P (1)

$$[c] = \begin{bmatrix} 200 & 150 & 180 & . & . & . & 90 & 120 & 50 & 150 & 80 \end{bmatrix}$$

Busses

$$\begin{bmatrix} 1 & 1 & 1 & . & . & . & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & . & . & . & 1 & 1 & 1 & 0 & 0 \\ . & . & . & . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & . & 0 & 0 & 0 & 1 & 1 \end{bmatrix} <=1$$

Trip segments

$$\begin{bmatrix} 1 & 1 & 0 & . & . & . & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & . & . & . & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & . & . & . & 1 & 0 & 0 & 1 & 1 \\ . & . & . & . & . & . & . & . & . & . & . \\ 1 & 0 & 0 & . & . & . & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & . & . & . & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & . & . & . & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & . & . & . & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & . & . & . & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & . & . & . & 0 & 0 & 0 & 1 & 0 \end{bmatrix} =1$$

Integer Program P (2)

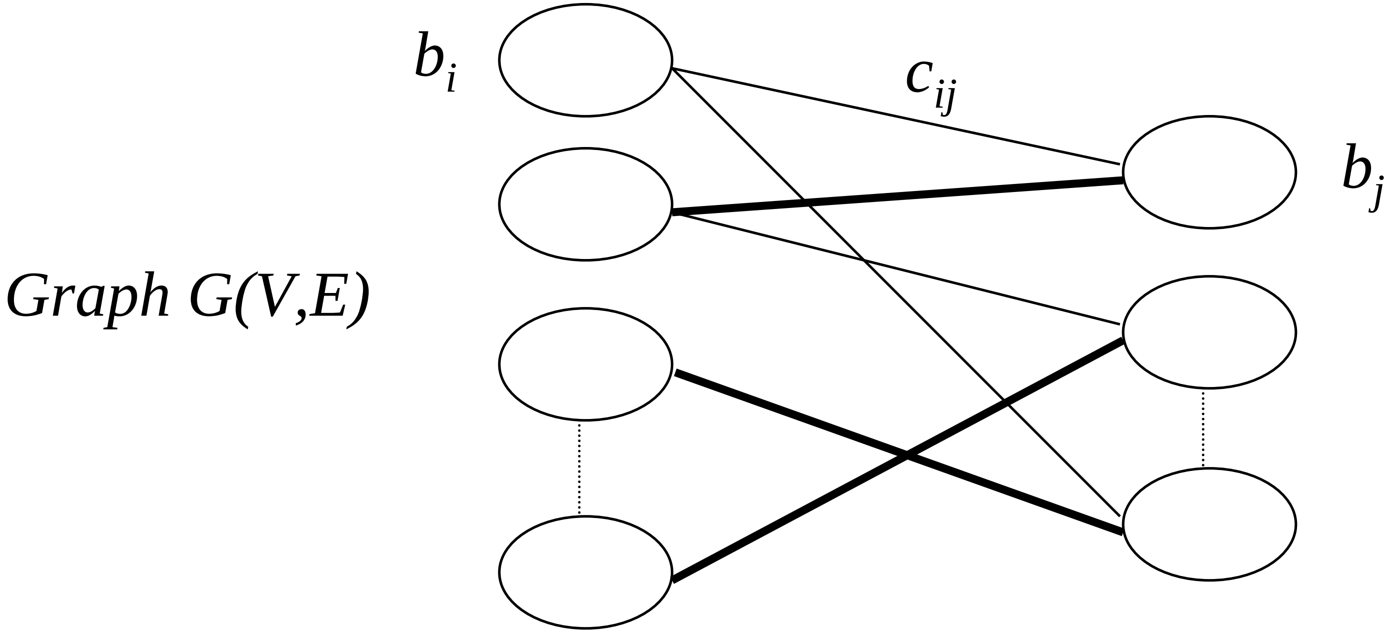
- Minimize $$\sum_{j=1}^{n} c_j x_j$$

- Subject to $$\sum_{j=1}^{n} a_{ij} x_j \leq 1$$

$$\sum_{j=1}^{n} b_{kj} x_j = 1 \qquad x_j = \{1,0\} \qquad i \in B, \quad k \in T$$

- *B* is the set of busses
- *T* is the set of trip segments
- *n* is the number of generated shifts
- c_j is the cost of a shift $j=1...n$

Minimum Cost Matching Problem

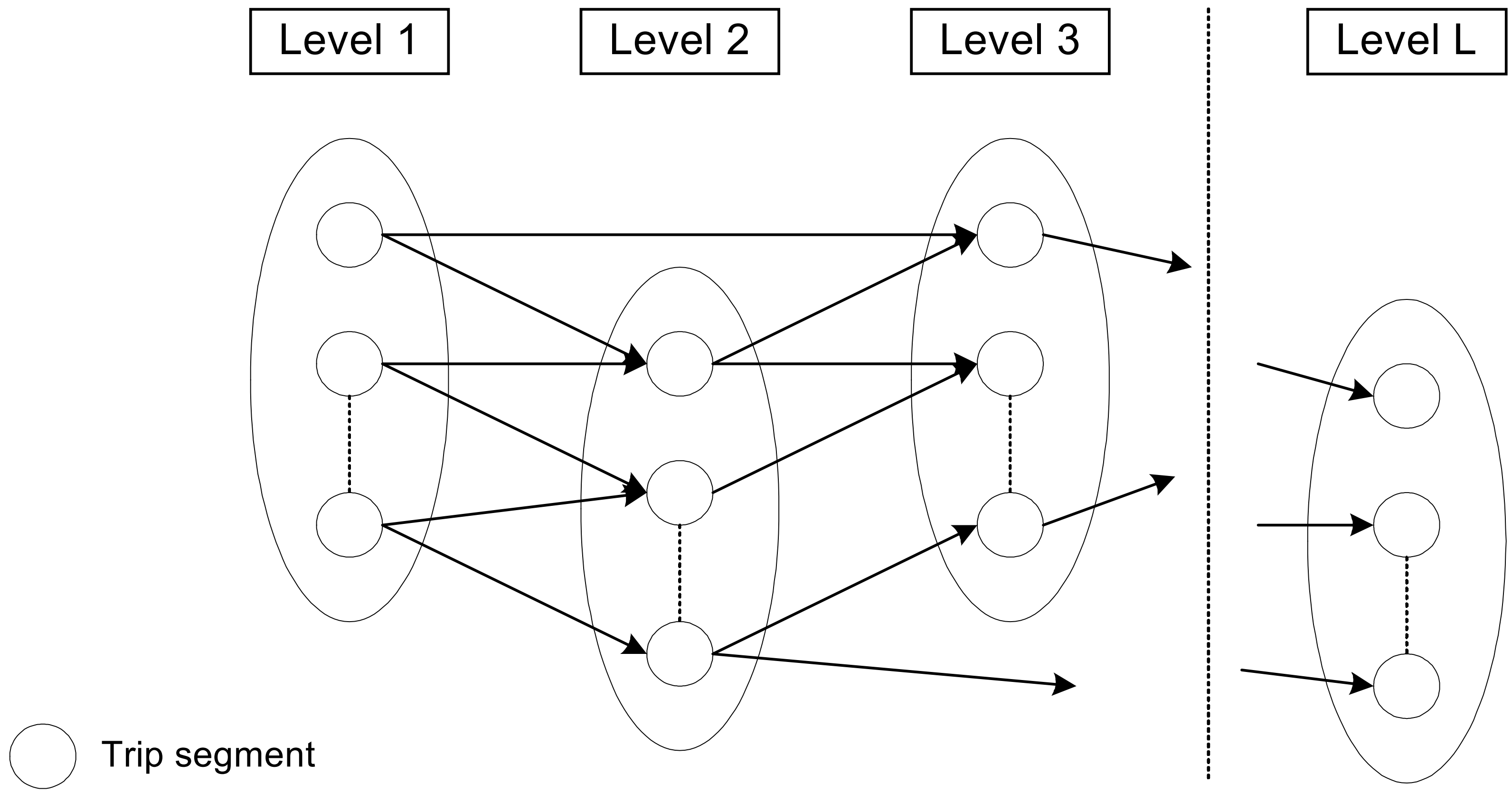


- Min $\{\sum_{(i,j)\in M} c_{ij} + \sum_{i\in(V-V1)} b_i\}$
- c_{ij} cost of edge $(i, j) \in E$
- b_i cost of vertex $i \in V$
- M is a matching
- $V1$ the set of vertices $\in M$

A Problem Specific Heuristic (QS)

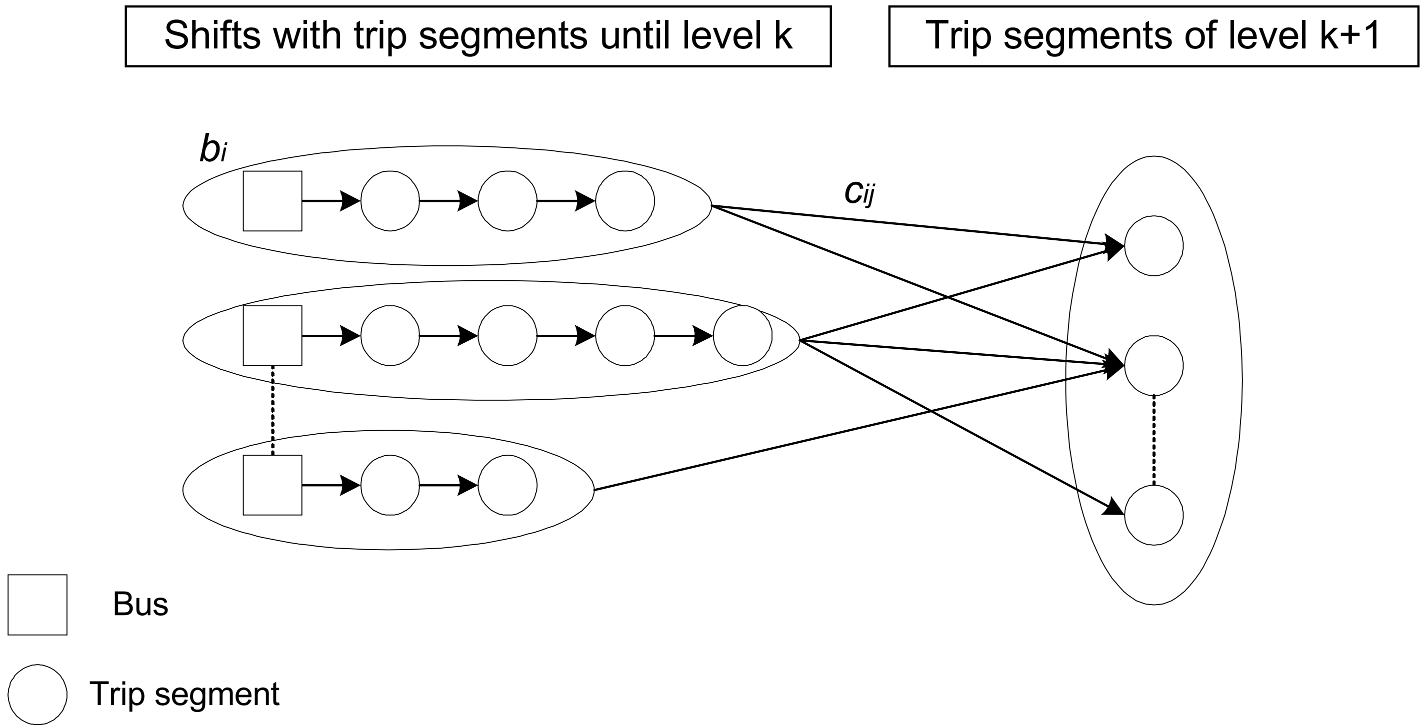
- Make initial shifts
  - Minimum cost matching
- Apply shifts improvement
  - 2-opt
  - 3-opt, Set Partitioning (SPP)
  - Shortest path

Partition of Trip Segments Into Levels



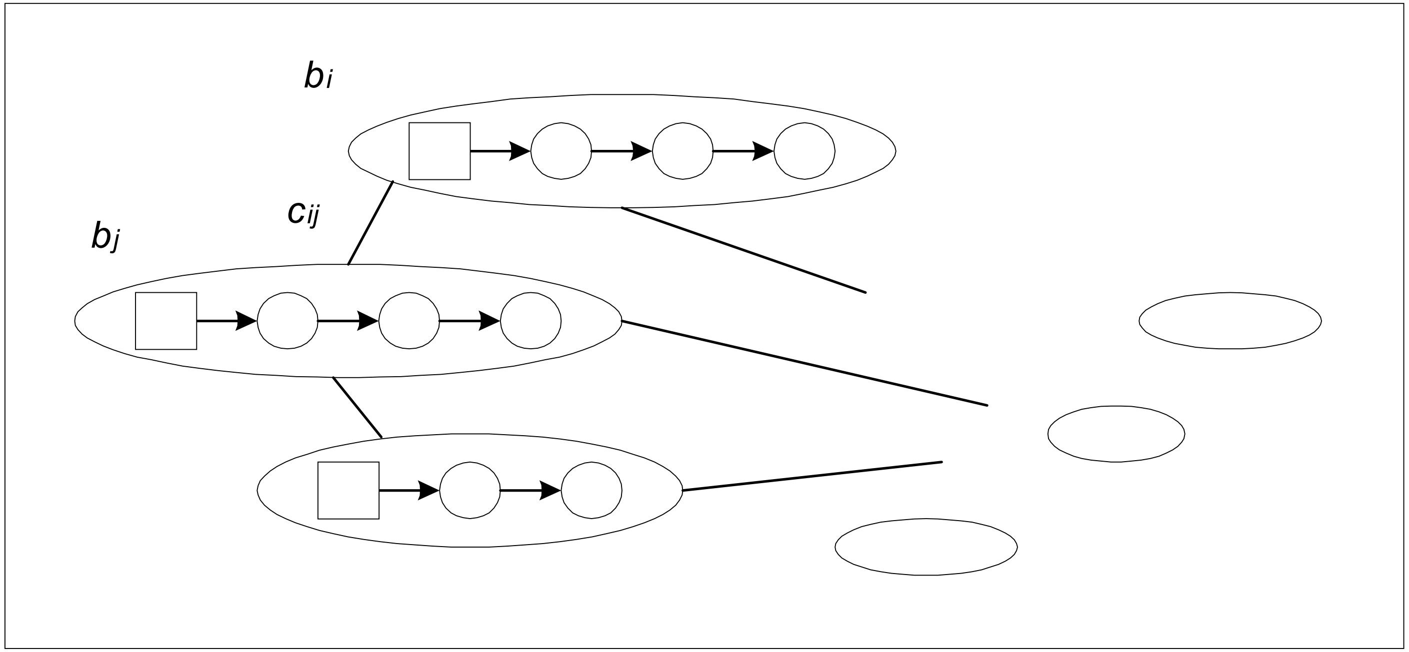
Initial Shifts

- Assign trip segments of level 1 to buses
- Enlargement of shifts with trip segments of level k+1



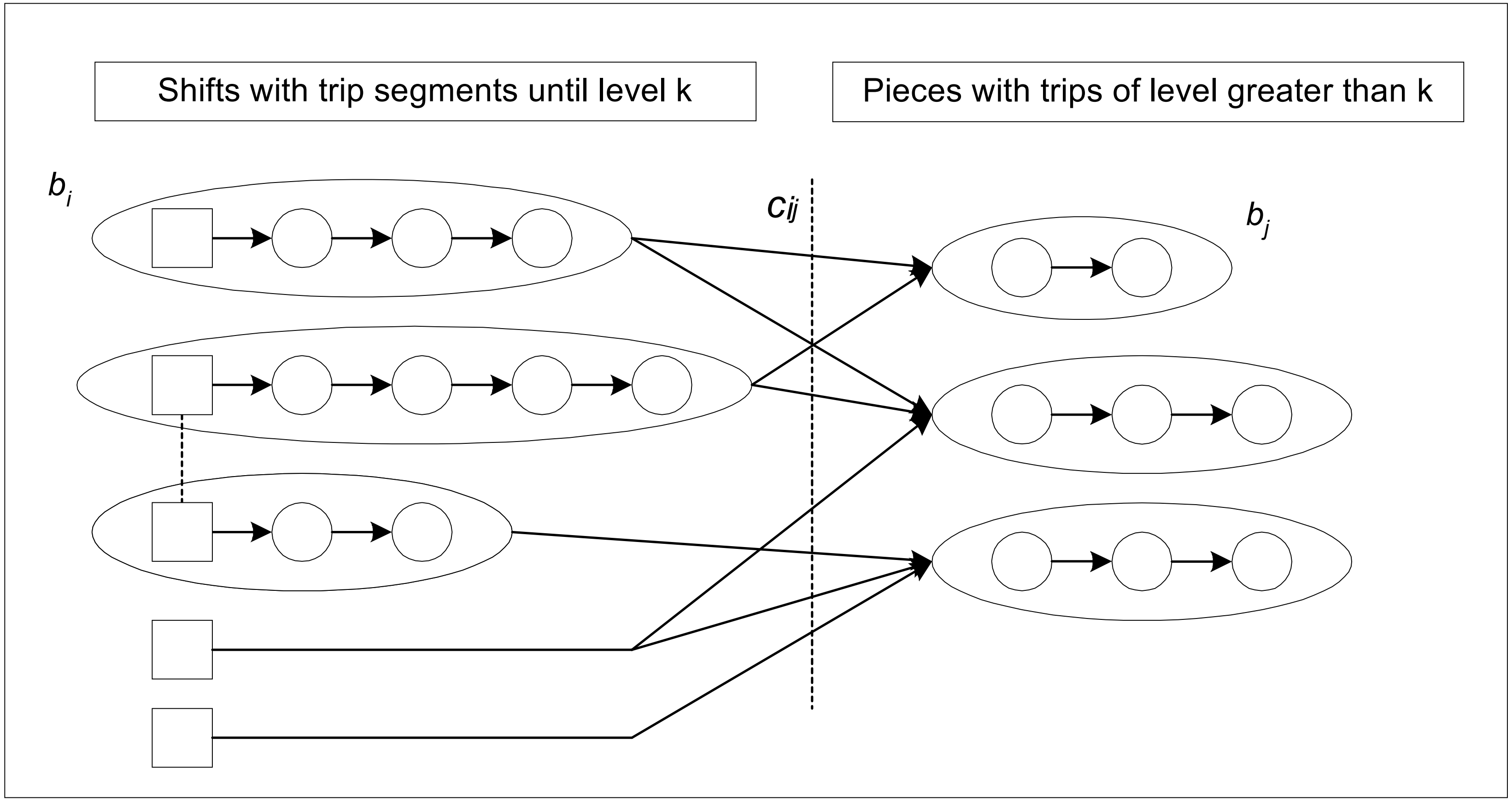
2-opt Improvement of Shifts

- Combine 2 shifts and find the best combinations to apply by solving a matching problem



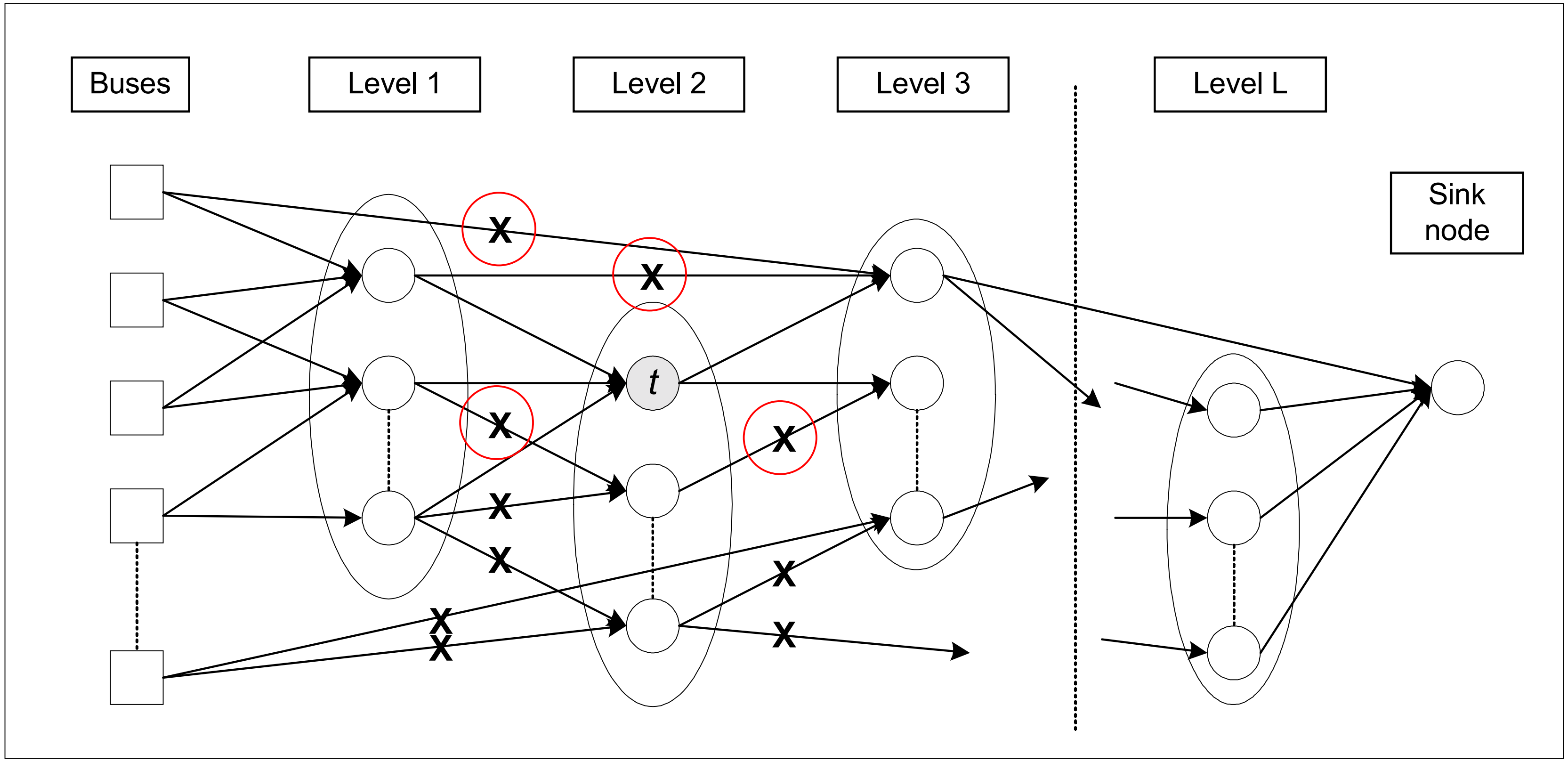
- b_i, b_j costs of shifts i, j
- c_{ij} the cost of the best interconnection between shifts i, j

Level Cutting Improvement of Shifts



Shifts for Uncovered Trip Segments

- k-shortest path



Quick Shift (QS) detailed Algorithm

- 1. Partition all trip segments into levels

For *a number of times*

- 2. Make initial shifts by matching the trip segments between consecutive levels

  Repeat

    Repeat

      - 3. Shifts improvement (2-opt search, level cutting)

    Until *no more improvements can be made*

    - 4. Shifts improvement 3-opt search (set partitioning)

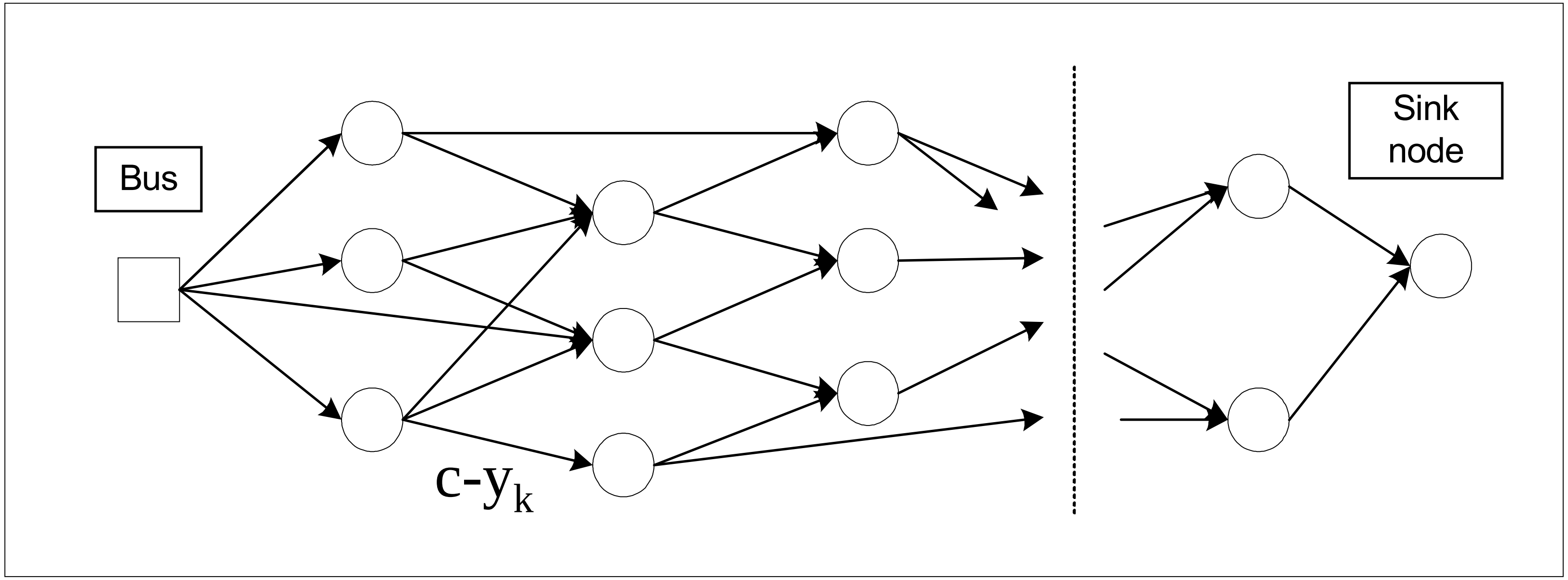
  Until *no more improvements can be made*

- 5. Find shifts for the uncovered trip segments (k-shortest path)
- 6. Keep some connections between trip segments from the solution

The Column Generation Algorithm (CGQS)

- **First Phase**
  - 1. Generate an initial solution (QS)
- **Second Phase**
  - **Repeat**
    - 2. Solve the linear relaxation of the complete model P
    - 3. Use QS to get an integer solution
    - 4. Generate additional columns in order to improve the quality of the linear program
  - **Until** stopping rules are satisfied

Generate Additional Columns (Shifts)

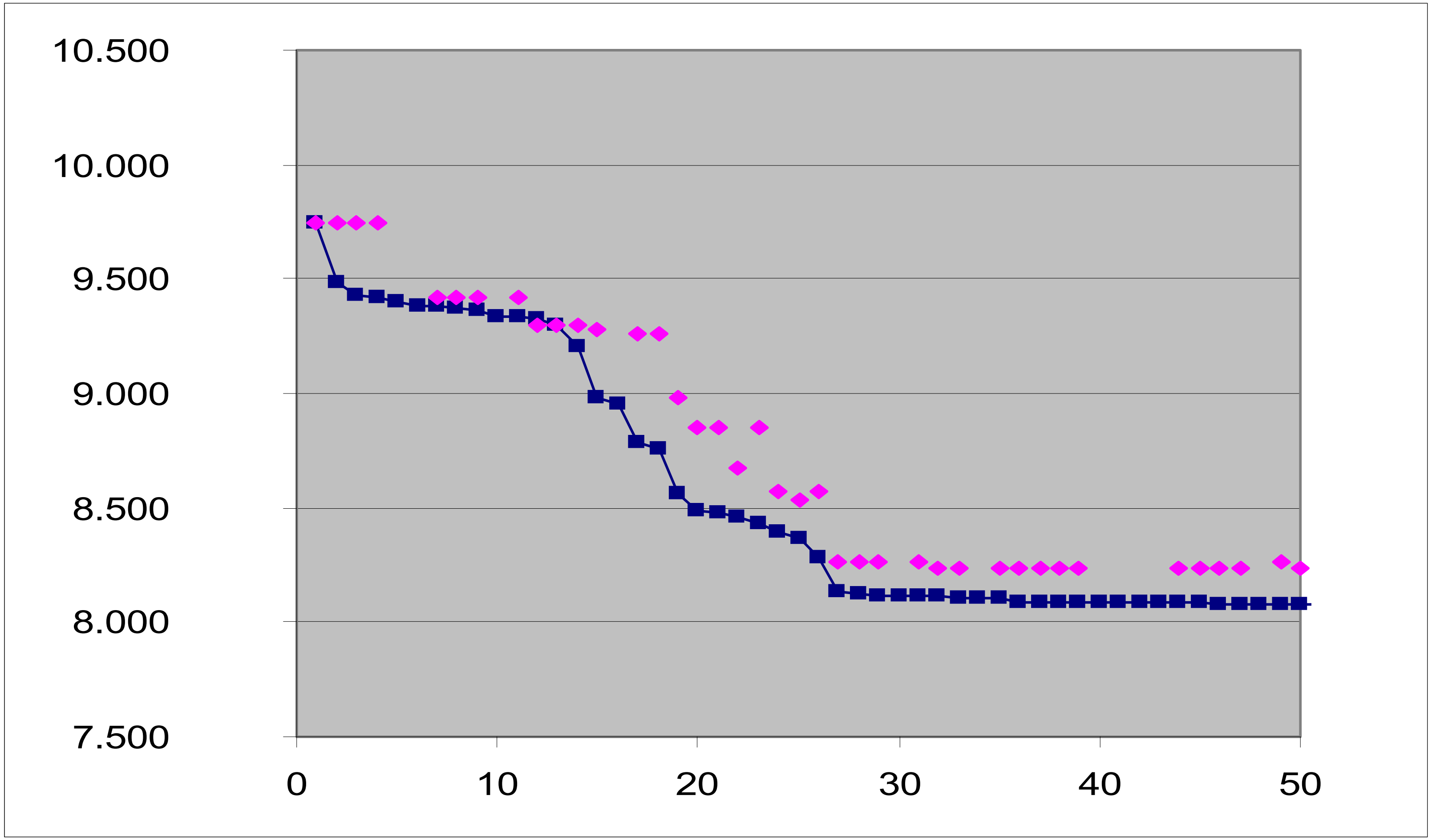


- Legal shifts
- Negative reduced cost $rc_j = c_j - y_i - \sum_{k \in S} y_k$
- DFS, Shortest paths

Computational Results

No. of busses	No. of trip segments	Total drive time (Min)	Idle time					
			QS		CGQS			LP
			Min	%	Min	%	%	Min
24	57	9855	105	1.1	105	1.1	0	105.00
46	134	21130	3210	13.2	2685	11.3	2.5	2617.20
60	158	30750	5055	14.1	4635	13.1	2.6	4515.97
48	175	26120	2890	10.0	1945	6.9	4.6	1857.88
58	264	20940	4160	16.6	3790	15.3	1.5	3734.16
89	337	34020	9745	22.3	8235	19.5	1.9	8077.44

Progress of the CGQS Algorithm



Conclusions

- QS is an application specific IP heuristic
  - Good solution (in production by itself!)
  - Extremely fast with minimal computer requirements (original design goal)
- CGQS a column generation approach
  - Significantly better solution
  - Use of the QS algorithm as the IP solver
  - Effective generation of additional shifts

Linear Program (1)

- Primal
  - Min cx
  - Subject to $Ax = b, x \geq 0$

- Dual
  - Max yb
  - Subject to $yA \leq c$

Linear Program (2)

- Duality Theorem
- Complementary Slackness Theorem
  - αν $x_j > 0 \Leftrightarrow y\,col_j(A) = c_j$
  - αν $x_j = 0 \Leftrightarrow y\,col_j(A) < c_j$
  - $\boldsymbol{rc_j = c_j - y\,col_j(A)}$
- Simplex, Interior Point

Δυναμική Δημιουργία Μεταβλητών (Column Generation)

Πρόβλημα επιλογής μεταβλητών

min cx
με τους περιορισμούς: Ax>=b
0<=x<=1

Πρόβλημα δημιουργίας μεταβλητών

- νόμιμες μεταβλητές
- μεταβλητές που βελτιώνουν την λύση του προβλήματος επιλογής

Πρόβλημα ακέραιας λύσης

- Εισαγωγή περιορισμών
- Ευρετικοί αλγόριθμοι
- Αλγόριθμοι ακέραιου προγρ/σμου

Υποσύστημα Ελέγχου Νομιμότητας

- Νόμιμη βάρδια, κόστος
- Ένα δρομολόγιο τμήμα βάρδιας λεωφορείου
- Δύο δρομολόγια συνεχόμενα σε βάρδια
- Ένα δρομολόγιο συνεχίζει μια βάρδια
- Επιθυμίες
  - Ανάθεση δρομολογίου σε οδηγό
  - Δύο δρομολόγια να είναι διαδοχικά

Τελική Επιλογή Λεωφορείων και Οδηγών

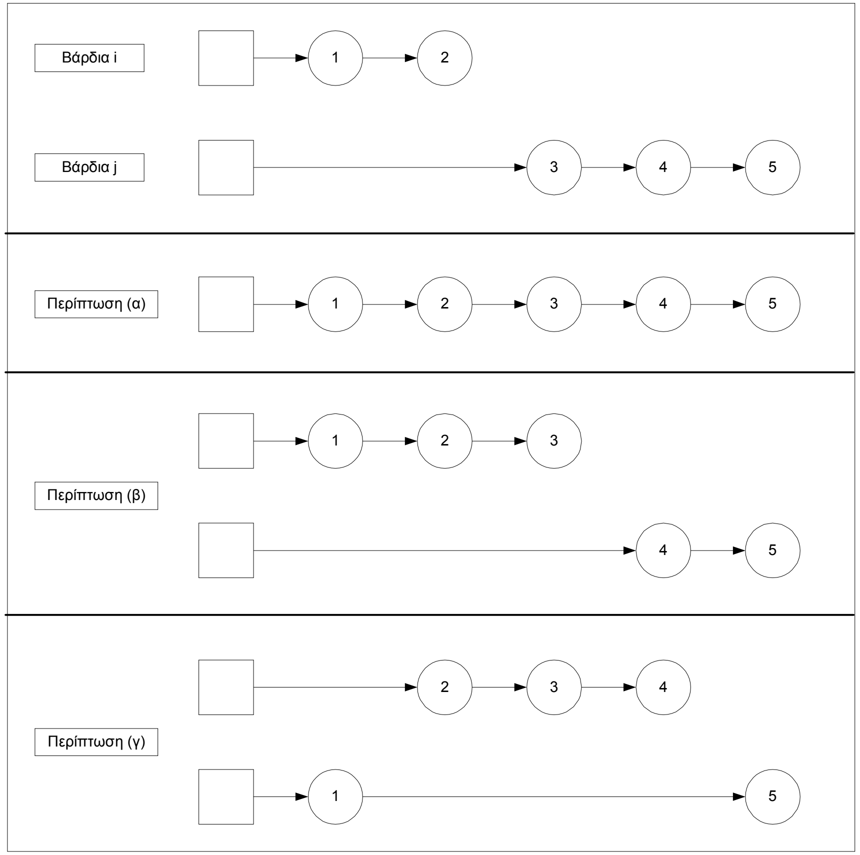
- Οικονομική και εφικτή λύση
  (ΔΛΟ ή ΔΛΟΔΔΜ)
- Ιστορικά στοιχεία για εξίσωση
  - υπερωριών
  - «εύκολων» και «δύσκολων» δρομολογίων
  - χιλιομέτρων
- Πρόβλημα «ταιριάσματος» με το ελάχιστο κόστος μεταξύ λεωφορείων-οδηγών και βαρδιών

Βελτίωση Βαρδιών ανά-3

- Συνδυασμός τριών βαρδιών και έλεγχος για αλλαγές
- Μικρό υποπρόβλημα, SPP
- Γραμμές λεωφορεία, δρομολόγια
- Στήλες νόμιμες νέες βάρδιες
- Επίλυση SPP (υπονοούμενη απαρίθμηση)
- Το συνολικό κόστος μικρότερο των αρχικών βαρδιών

Βελτίωση Βαρδιών ανά-2 (*πρώτη φάση*)

- Συνδυασμός δύο βαρδιών και έλεγχος για αλλαγές



(1)

(2)

(3)

- νόμιμες νέες βάρδιες
- συνολικό κόστος μικρότερο των αρχικών

Επιλογή Συνδέσεων Δρομολογίων από τη λύση

- Αρχική λύση καθορίζει την ποιότητα της τελικής λύσης
- Επιλογή συνδέσεων δρομολογίων από τις καλύτερες βάρδιες
- Κατά τη δημιουργία αρχικών βαρδιών αξιοποιείται πληροφορία για καλή λύση