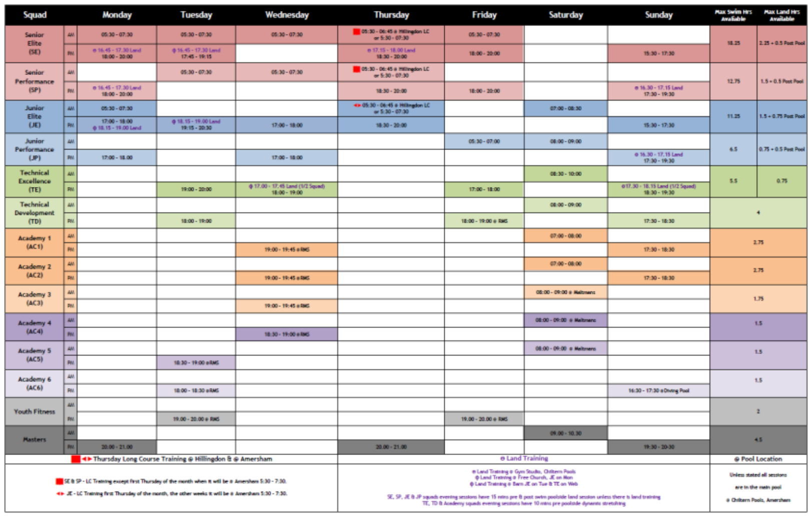
Two Example Optimisation Problems from the World of Education



Squad		Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Max Swim Hrs Available	Max Land Hrs Available
Senior Elite (SE)	AM	05:30 - 07:30	05:30 - 07:30	05:30 - 07:30	🟥 05:30 - 06:45 @ Hillingdon LC or 5:30 - 07:30	05:30 - 07:30			18.25	2.25 + 0.5 Post Pool
	PM	⊕ 16.45 - 17.30 Land 18:00 - 20:00	ɸ 16.45 - 17.30 Land 17:45 - 19:15		⊕ 17.15 - 18.00 Land 18:30 - 20:00	18:00 - 20:00		15:30 - 17:30		
Senior Performance (SP)	AM		05:30 - 07:30	05:30 - 07:30	🟥 05:30 - 06:45 @ Hillingdon LC or 5:30 - 07:30				12.75	1.5 + 0.5 Post Pool
	PM	⊕ 16.45 - 17.30 Land 18:00 - 20:00			18:30 - 20:00	18:00 - 20:00		⊕ 16.30 - 17.15 Land 17:30 - 19:30		
Junior Elite (JE)	AM	05:30 - 07:30			◂▸ 05:30 - 06:45 @ Hillingdon LC or 5:30 - 07:30		07:00 - 08:30		11.25	1.5 + 0.75 Post Pool
	PM	17:00 - 18:00 ɸ 18.15 - 19.00 Land	ɸ 18.15 - 19.00 Land 19:15 - 20:30	17:00 - 18:00	18:30 - 20:00			15:30 - 17:30		
Junior Performance (JP)	AM					05:30 - 07:00	08:00 - 09:00		6.5	0.75 + 0.5 Post Pool
	PM	17:00 - 18.00		17:00 - 18:00				⊕ 16.30 - 17.15 Land 17:30 - 19:30		
Technical Excellence (TE)	AM						08:30 - 10:00		5.5	0.75
	PM		19:00 - 20:00	ɸ 17.00 - 17.45 Land (1/2 Squad) 18:00 - 19:00		17:00 - 18:00		⊕17.30 - 18.15 Land (1/2 Squad) 18:30 - 19:30		
Technical Development (TD)	AM						08:00 - 09:00		4	
	PM		18:00 - 19:00			18:00 - 19:00 @ RMS		17:30 - 18:30		
Academy 1 (AC1)	AM						07:00 - 08:00		2.75	
	PM			19:00 - 19:45 @RMS				17:30 - 18:30		
Academy 2 (AC2)	AM						07:00 - 08:00		2.75	
	PM			19:00 - 19:45 @RMS				17:30 - 18:30		
Academy 3 (AC3)	AM						08:00 - 09:00 @ Maltmans		1.75	
	PM			19:00 - 19:45 @RMS						
Academy 4 (AC4)	AM						08:00 - 09:00 @ Maltmans		1.5	
	PM			18:30 - 19:00 @RMS						
Academy 5 (AC5)	AM						08:00 - 09:00 @ Maltmans		1.5	
	PM		18:30 - 19:00 @RMS							
Academy 6 (AC6)	AM								1.5	
	PM		18:00 - 18:30 @RMS					16:30 - 17:30 @Diving Pool		
Youth Fitness	AM								2	
	PM		19.00 - 20.00 @ RMS			19.00 - 20.00 @ RMS				
Masters	AM						09.00 - 10.30		4.5	
	PM	20.00 - 21.00			20.00 - 21.00			19:30 - 20:30		

🟥 ◂▸ Thursday Long Course Training @ Hillingdon & @ Amersham

🟥 SE & SP - LC Training except first Thursday of the month when it will be @ Amersham 5:30 - 7:30.

◂▸ JE - LC Training first Thursday of the month, the other weeks it will be @ Amersham 5:30 - 7:30.

⊕ Land Training

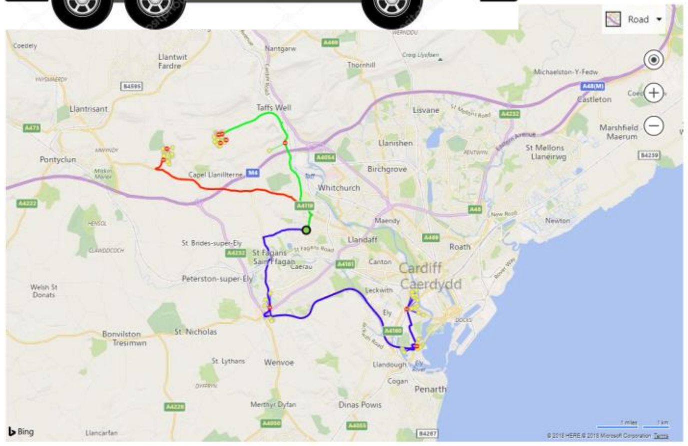
⊕ Land Training @ Gym Studio, Chiltern Pools
ɸ Land Training @ Free Church, JE on Mon
ɸ Land Training @ Barn JE on Tue & TE on Wed

SE, SP, JE & JP squads evening sessions have 15 mins pre & post swim poolside land session unless there is land training
TE, TD & Academy squads evening sessions have 10 mins pre poolside dynamic stretching

@ Pool Location

Unless stated all sessions are in the main pool

@ Chiltern Pools, Amersham



Lewis, R. and J. Thompson (2015) 'Analysing the Effects of Solution Space Connectivity with an Effective Metaheuristic for the Course Timetabling Problem'. *European Journal of Operational Research*, vol. 240, pp. 637-648.

Lewis, R., K. Smith-Miles, and K. Phillips (2018) 'The School Bus Routing Problem: An Analysis and Algorithm'. In *Combinatorial Algorithms* (LNCS 10765), Springer, pp. 287-298.

Rhyd Lewis

School of Mathematics, Cardiff University,

LewisR9@cf.ac.uk, www.RhydLewis.eu

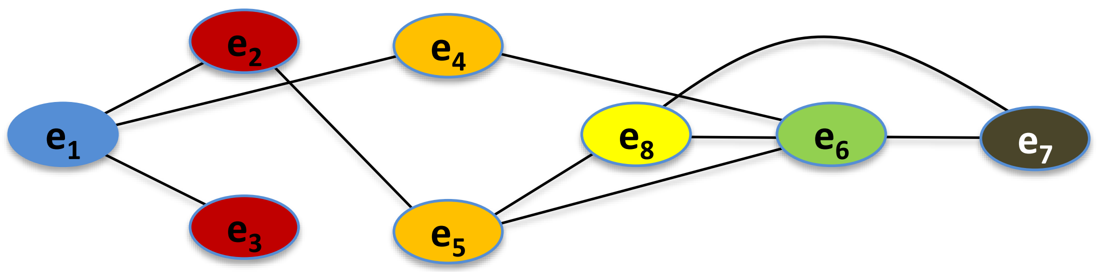
Problem I: Post Enrollment Timetabling

- This problem has been the subject of a few international competitions
- We need to assign a set of “events” (lectures, etc.) to “timeslots” and rooms
- Each event has a list of attending students
- Hard Constraints
  - No double booking of rooms or students
  - Some events should occur before / after others
  - Some timeslots are forbidden for certain events
  - Events should only be assigned to suitable rooms with adequate seating

Time-Slots	1	2	3	4	5	6	7	8	…	45
Room 1	e_1		e_5							
Room 2		e_2	e_4			e_6		e_7		
Room 3		e_3		e_8						
Room 4										

Relationship to Graph Colouring

- The problem generalises a graph colouring problem



= "cannot be put into the same timeslot" (e.g. student clash)

Time-Slots	1	2	3	4	5	6	7	8	…	45
Room 1	e_1		e_5							
Room 2		e_2	e_4			e_6		e_7		
Room 3		e_3		e_8						
Room 4										

Stage 1: Finding Feasibility

Strategy:

- Using heuristics, insert as many events as possible into the timetable such that the hard constraints are obeyed.
- Keep any remaining events in a list U.
- Now make adjustments to the timetable so that U is emptied. This gives a full feasible timetable.

Time-Slots	1	2	3	4	5	6	7	8	…	45
Room 1	e_1		e_5							
Room 2		e_2	e_4			e_6		e_7		
Room 3		e_3		e_8						
Room 4										

$U = \{e_9, e_{10}, e_{11}\}$

Stage 1: Finding Feasibility

Strategy:

- Using heuristics, insert as many events as possible into the timetable such that the hard constraints are obeyed.
- Keep any remaining events in a list U.
- Now make adjustments to the timetable so that U is emptied. This gives a full feasible timetable.

Time-Slots	1	2	3	4	5	6	7	8	…	45
Room 1	e_1		e_5							
Room 2		e_2	e_4			e_6		e_7		
Room 3		e_3		e_8						
Room 4										

$U = \{e_9, e_{10}, e_{11}\}$

Stage 1: Finding Feasibility

Strategy:

- Using heuristics, insert as many events as possible into the timetable such that the hard constraints are obeyed.
- Keep any remaining events in a list U.
- Now make adjustments to the timetable so that U is emptied. This gives a full feasible timetable.

Time-Slots	1	2	3	4	5	6	7	8	…	45
Room 1	e_1		e_5							
Room 2		e_2	e_4			e_6		e_7		
Room 3		e_3		e_8						
Room 4		e_{11}								

$U = \{e_9, e_{10} \quad \}$

Stage 1: Finding Feasibility

Strategy:

- Using heuristics, insert as many events as possible into the timetable such that the hard constraints are obeyed.
- Keep any remaining events in a list U.
- Now make adjustments to the timetable so that U is emptied. This gives a full feasible timetable.

Time-Slots	1	2	3	4	5	6	7	8	…	45
Room 1	e_1		e_5							
Room 2		e_2	e_4			e_6		e_7		
Room 3		e_3		e_8						
Room 4		e[illegible]								

U = {e_9, e_{10} }

Stage 1: Finding Feasibility

Strategy:

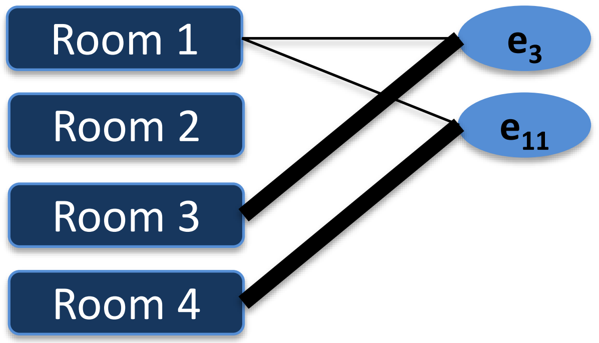
- Using heuristics, insert as many events as possible into the timetable such that the hard constraints are obeyed.
- Keep any remaining events in a list U.
- Now make adjustments to the timetable so that U is emptied. This gives a full feasible timetable.

Time-Slots	1	2	3	4	5	6	7	8	…	45
Room 1	e_1		e_5							
Room 2			e_4			e_6		e_7		
Room 3		e_3								
Room 4		e_{11}								

$U = \{e_9, e_{10}, e_2, e_8\}$

Room Allocations via maximum matching

- Extra flexibility is also offered if we treat room allocation as a maximum bipartite matching problem

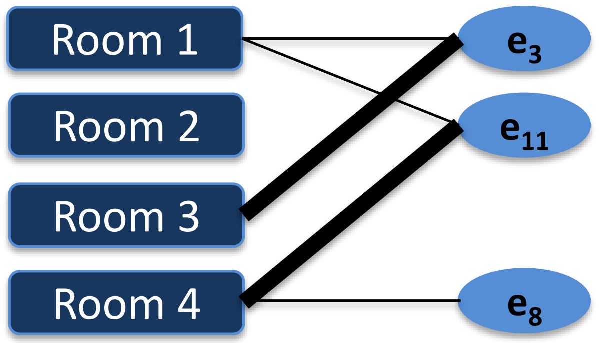


Time-Slots	1	2	3	4	5	6	7	8	…	45
Room 1	e_1		e_5							
Room 2			e_4			e_6		e_7		
Room 3		e_3								
Room 4		e_{11}								

$U = \{e_9, e_{10}, e_2, e_8\}$

Room Allocations via maximum matching

- Extra flexibility is also offered if we treat room allocation as a maximum bipartite matching problem

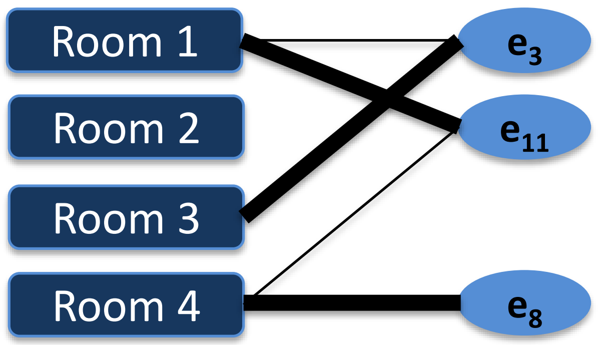


Time-Slots	1	2	3	4	5	6	7	8	…	45
Room 1	e_1		e_5							
Room 2			e_4			e_6		e_7		
Room 3		e_3								
Room 4		e_{11}								

$U = \{e_9, e_{10}, e_2, e_8\}$

Room Allocations via maximum matching

- Extra flexibility is also offered if we treat room allocation as a maximum bipartite matching problem



Time-Slots	1	2	3	4	5	6	7	8	…	45
Room 1	e_1	e_{11}	e_5							
Room 2			e_4			e_6		e_7		
Room 3		e_3								
Room 4		e_8								

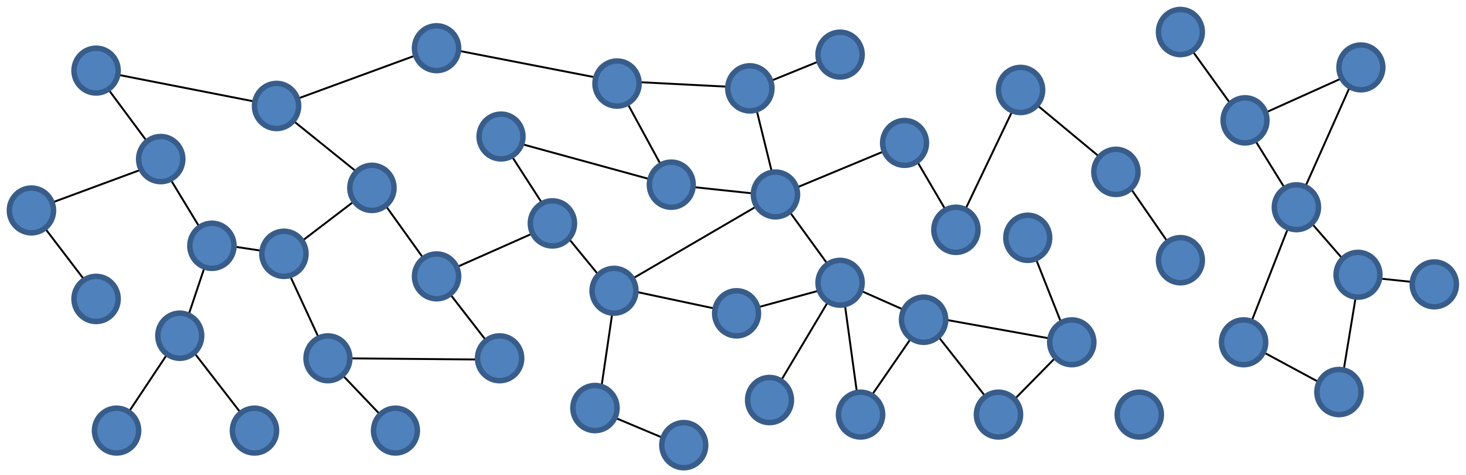
$U = \{e_9, e_{10}, e_2\}$

Stage 2: Eliminating Soft Constraint Violations

Strategy:

- Once feasibility is achieved, we now explore the space of feasible solutions, seeking to minimise a cost that reflects the number of soft constraint violations.

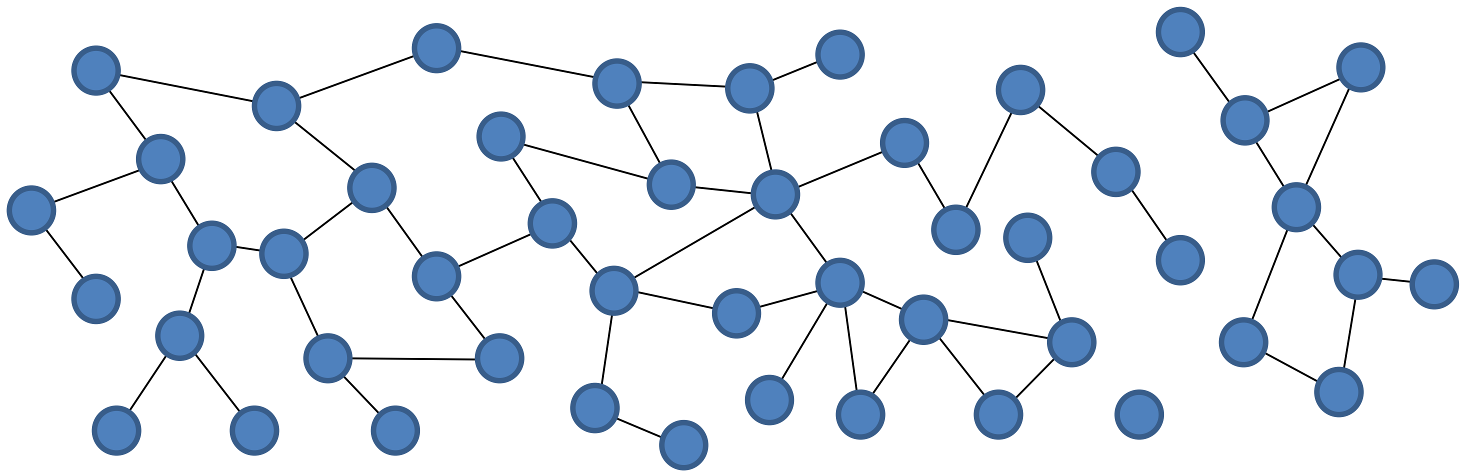
- This can be achieved by applying neighborhood moves, but rejecting them if they violate a hard constraint



Space of all feasible solutions. Edges indicate the existence of a neighbourhood move from one solution to another

Feasibility Ratio

- The connectivity of the solution space is very important, though it is usually too large to formally measure
- An indication can be gained using the **Feasibility Ratio**, which is the proportion of tested neighbourhood moves that are seen to retain feasibility (whether accepted or not).



Space of all feasible solutions. Edges indicate the existence of a neighbourhood move from one solution to another

Neighbourhood Operators

- Five neighbourhood operators $N_1,\ldots,N_5$ were designed. Each one is an extension of the previous one and should therefore increase the feasibility ratio:

- **N_1**: Choose an event and move it to a new timeslot
  OR
  Choose two events and swap their timeslots

- **N_2**: As with N_1, but apply a maximum matching algorithm to reallocate rooms if necessary.

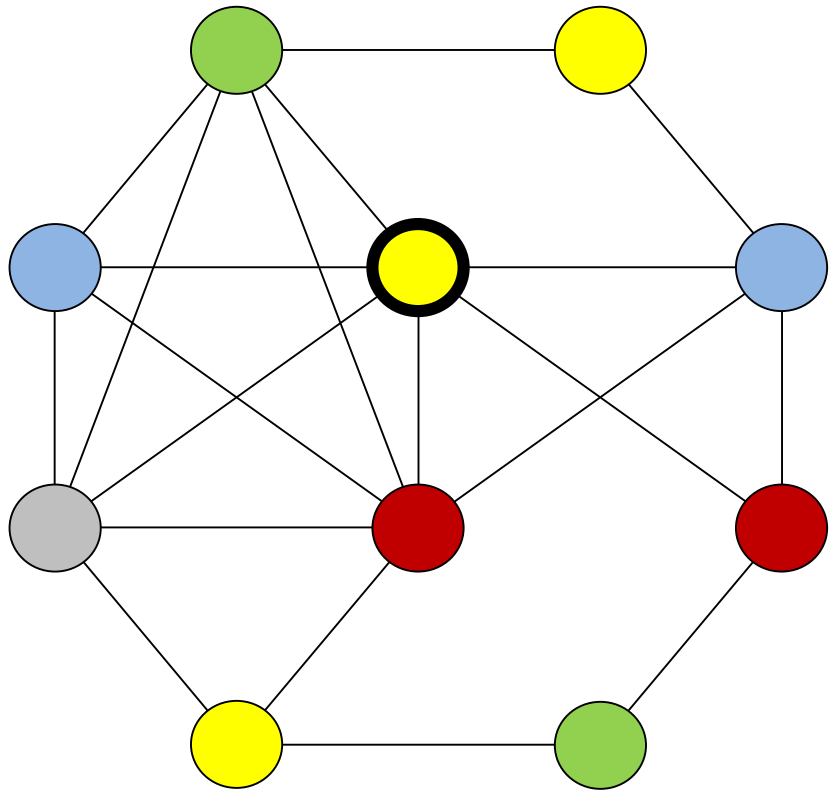
Time-Slots	1	2	3	4	5	6	7	8	…	45
Room 1	e_1	e_{11}	e_5	e_9						
Room 2			e_4			e_6		e_7		
Room 3		e_3		e_2	e_{10}					
Room 4		e_8								

Kempe Chains in Graph Colouring

Given a feasible colouring,

- Take a vertex of colour i and a different colour j.
- Form a connected subgraph containing this vertex and any others with colours i and j.
- Swap the colours of the vertices

This interchange of colours is guaranteed to retain feasibility.

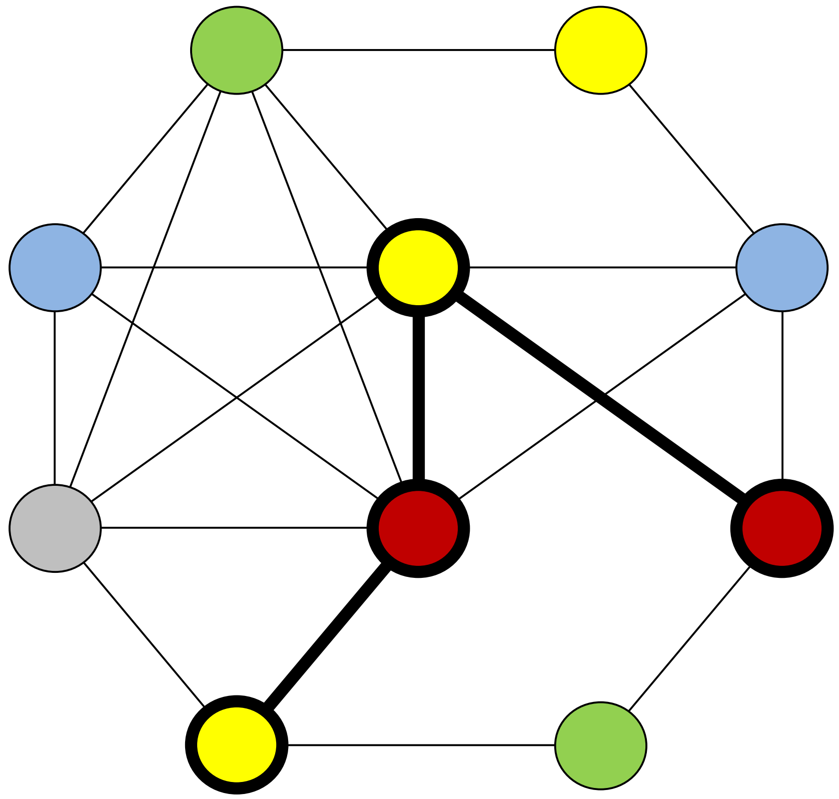


Kempe Chains in Graph Colouring

Given a feasible colouring,

- Take a vertex of colour i and a different colour j.
- Form a connected subgraph containing this vertex and any others with colours i and j.
- Swap the colours of the vertices

This interchange of colours is guaranteed to retain feasibility.

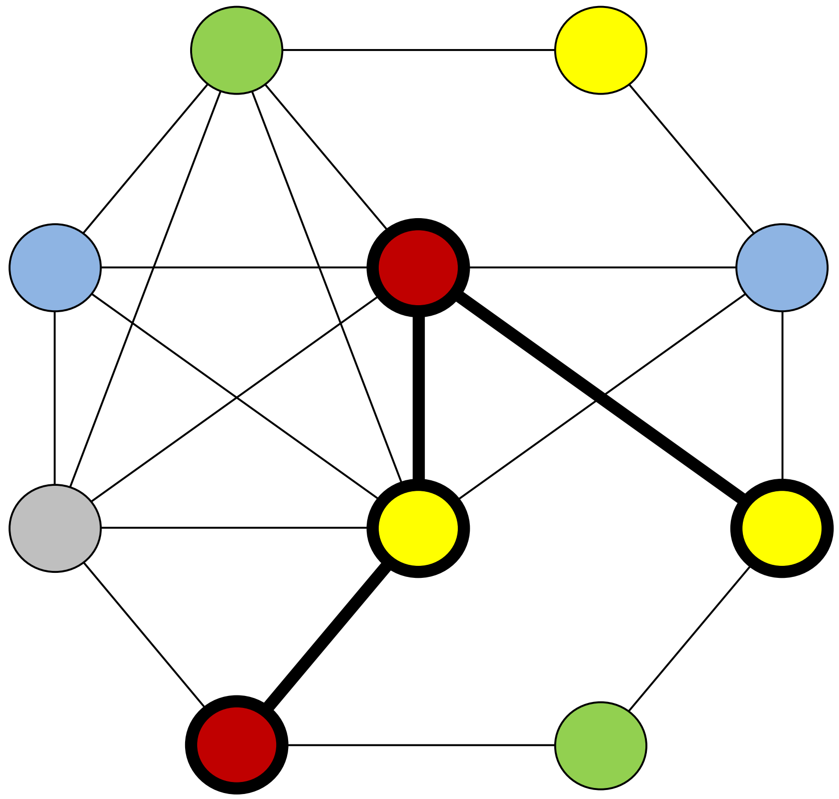


Kempe Chains in Graph Colouring

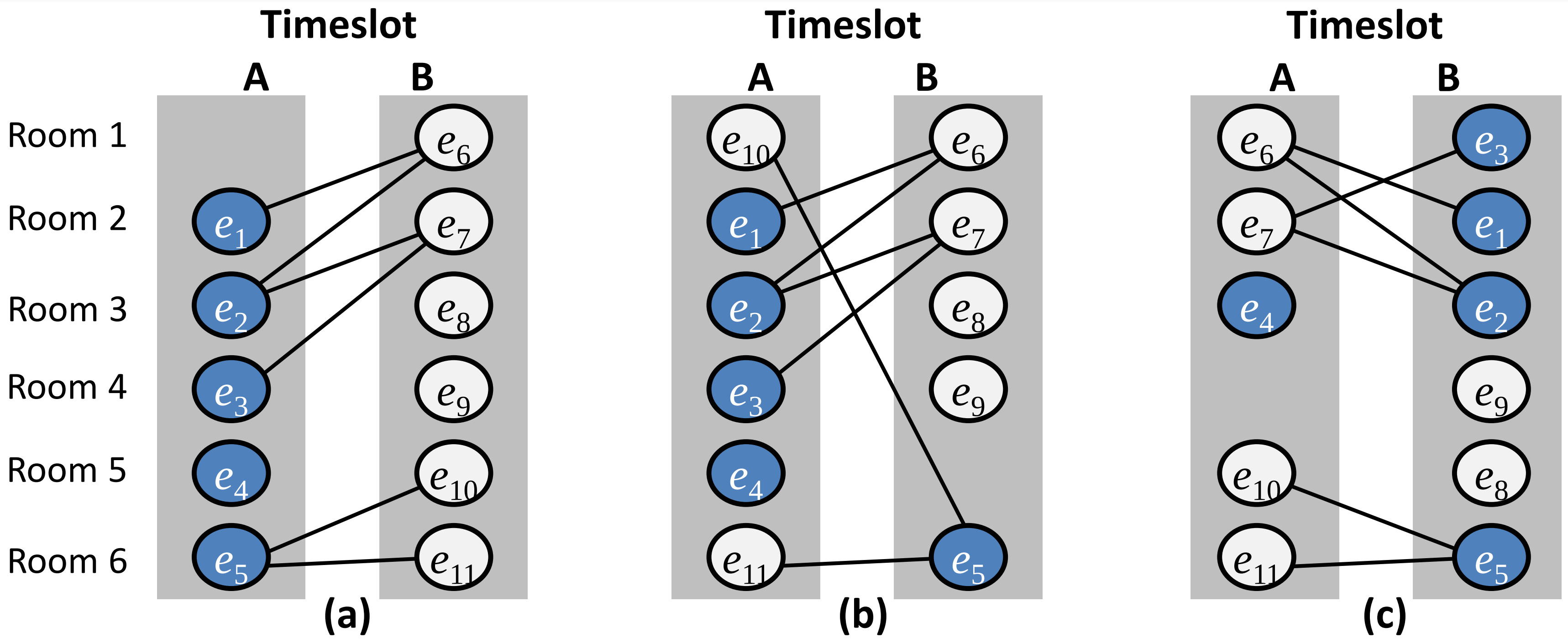
Given a feasible colouring,

- Take a vertex of colour i and a different colour j.
- Form a connected subgraph containing this vertex and any others with colours i and j.
- Swap the colours of the vertices

This interchange of colours is guaranteed to retain feasibility.



More Neighbourhood Operators…


Timeslot
A
B
Room 1
Room 2
Room 3
Room 4
Room 5
Room 6
e_1 e_2 e_3 e_4 e_5 e_6 e_7 e_8 e_9 e_{10} e_{11}
(a)
(b)
(c)

N$_3$: Perform a Kempe chain interchange, and use a maximum matching algorithm for room allocations. ((a) and (b) above).

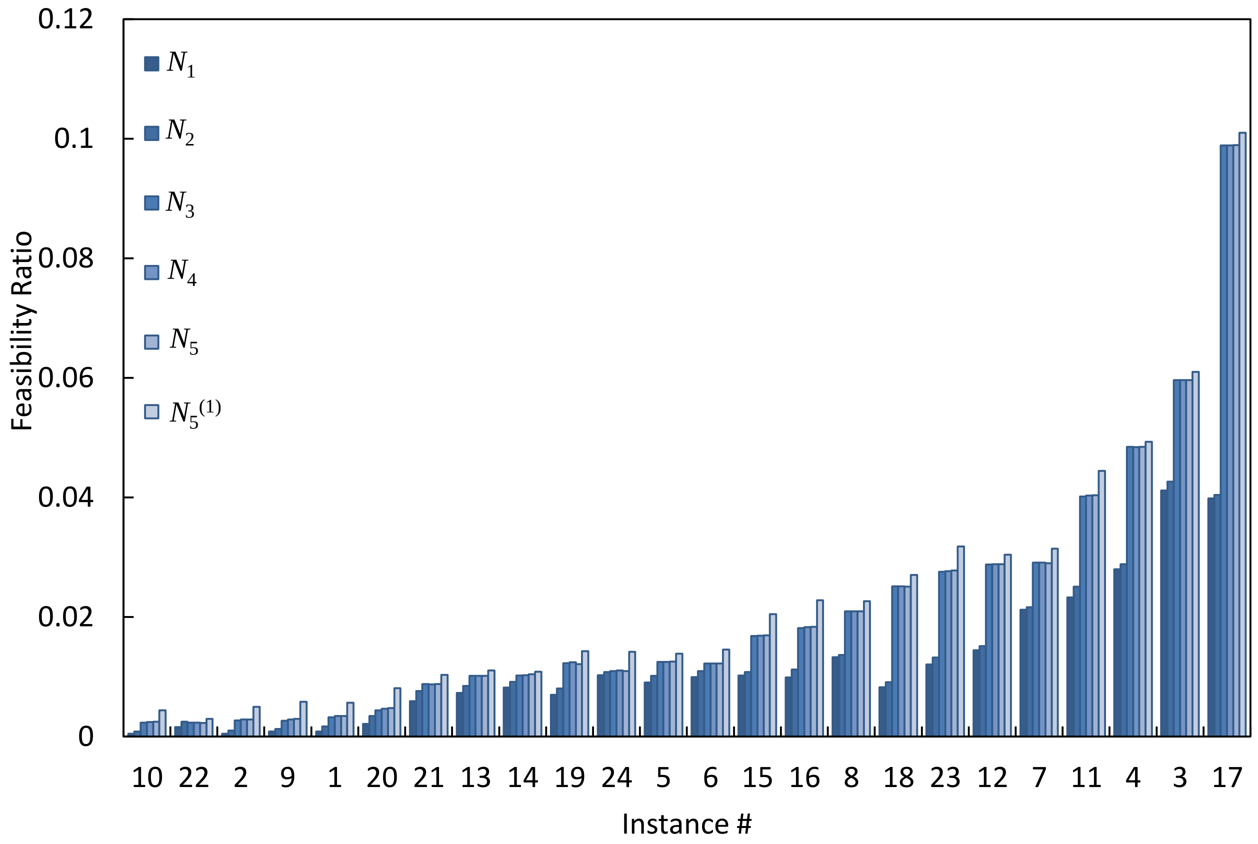
N$_4$ and N$_5$: As with N$_3$, but perform multiple Kempe chain interchanges if a single Kempe chain is seen to violate the constraints regarding room allocation ((a) and (c) above).

Yet More Neighbourhood Operators…

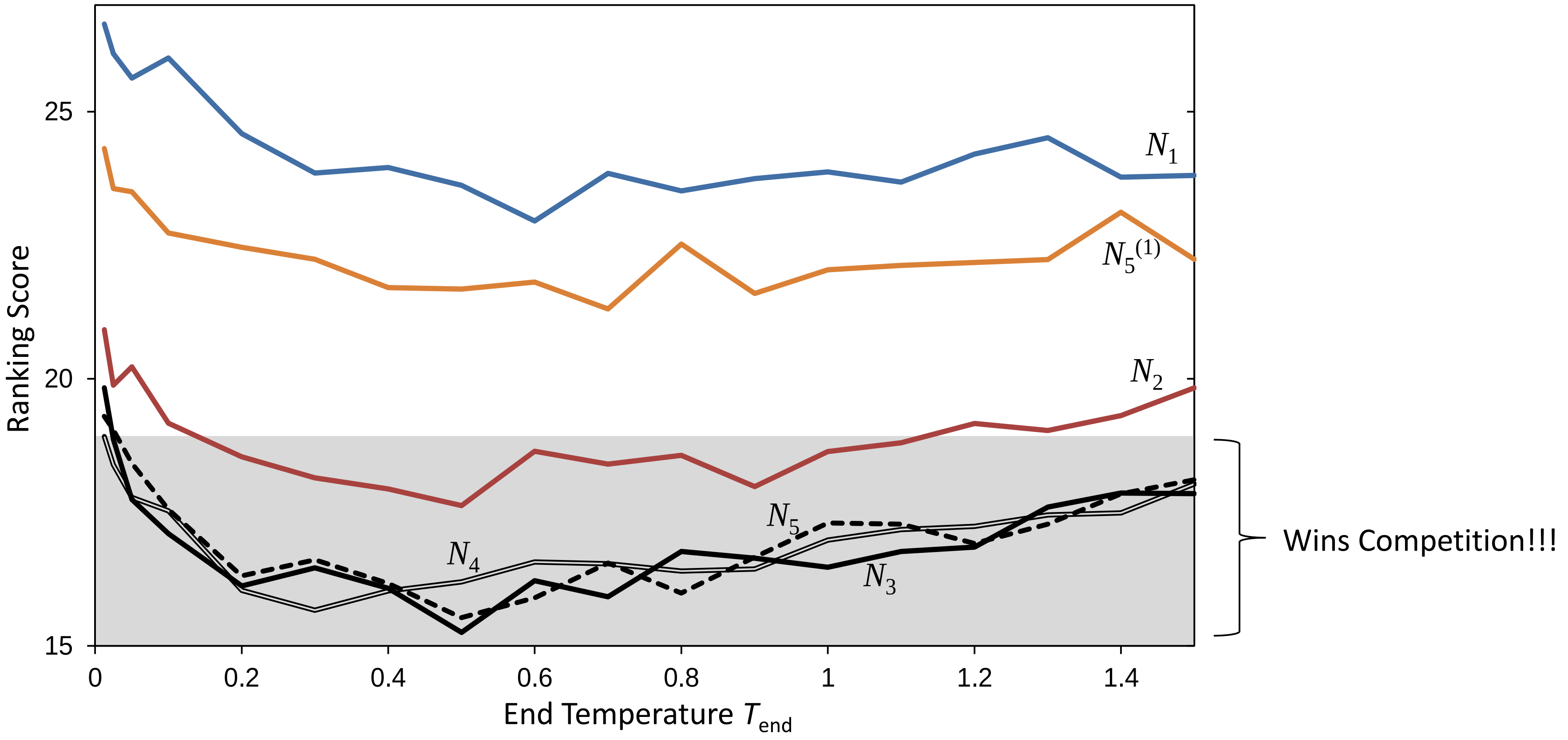
- We can also create additional dummy rooms to increase the feasibility ratio.
- However, use of dummy rooms must be discouraged via additional penalties in the cost function.
- A neighbourhood operator used with x dummy rooms is denoted $\mathbf{N_i^{(x)}}$

Time-Slots	1	2	3	4	5	6	7	8	…	45
Room 1	e_1	e_{11}	e_5	e_9						
Room 2			e_4			e_6		e_7		
Room 3					e_{10}					
Room 4		e_8								
Dummy Room				e_2						
Dummy Room				e_3						

Feasibility Ratios in Available Problem Instances

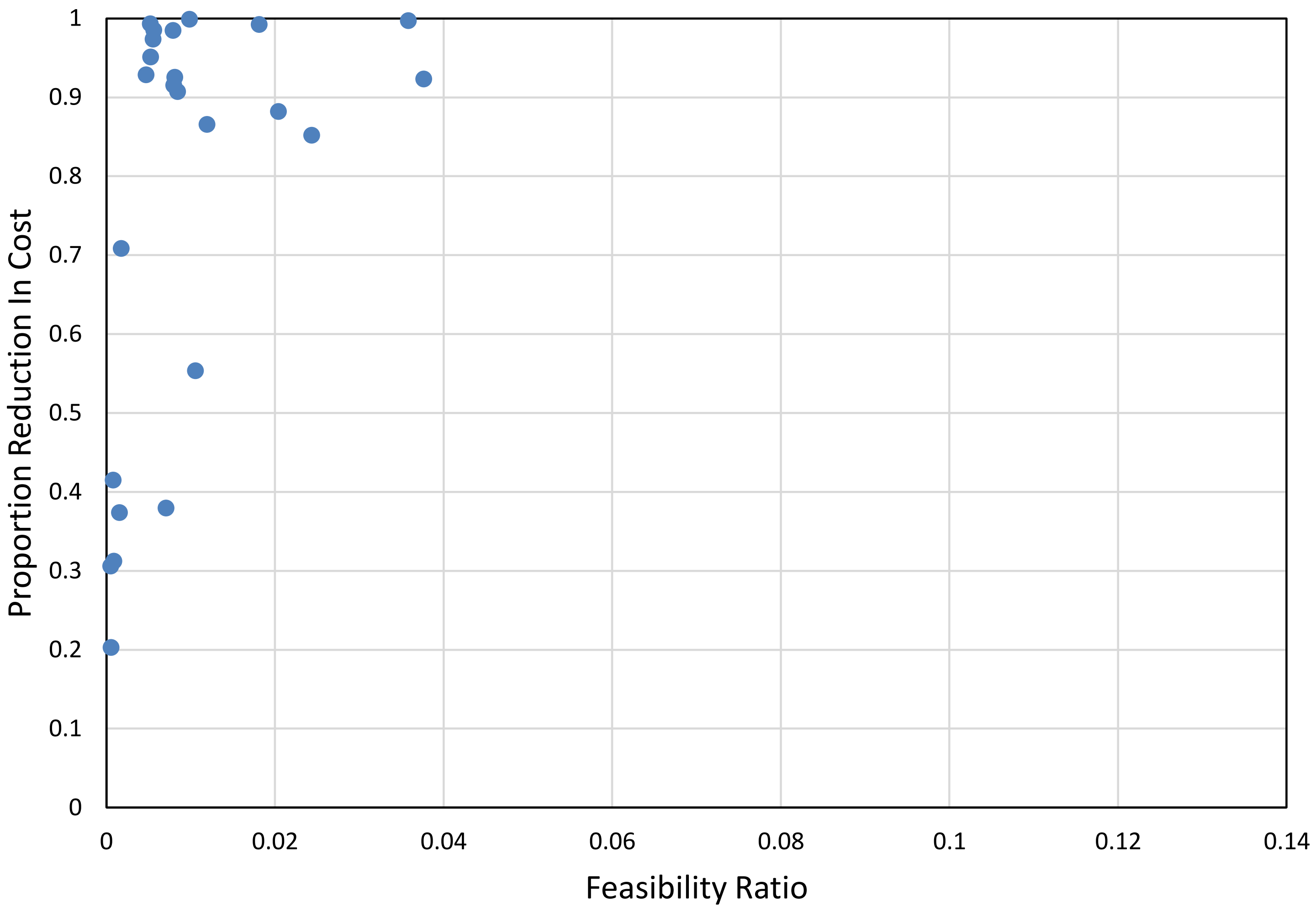


Feasibility ratios for different neighbourhood operators for all 24 problem instances used in the 2007 International Timetabling Competition. (Taken from random walks in the solution space)

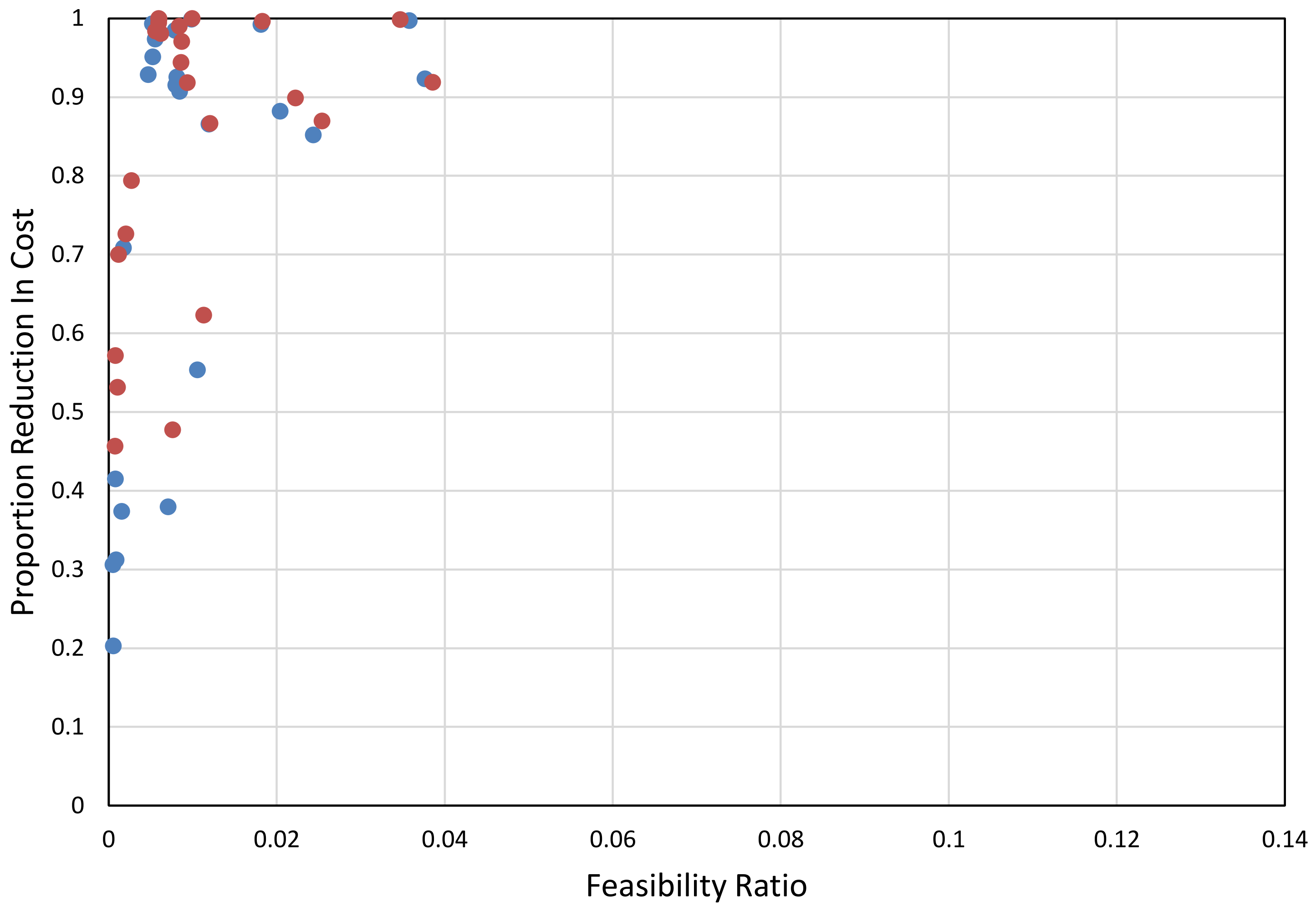


Performance of a simulated annealing algorithm (using different end temperatures) with the various neighbourhood operators.

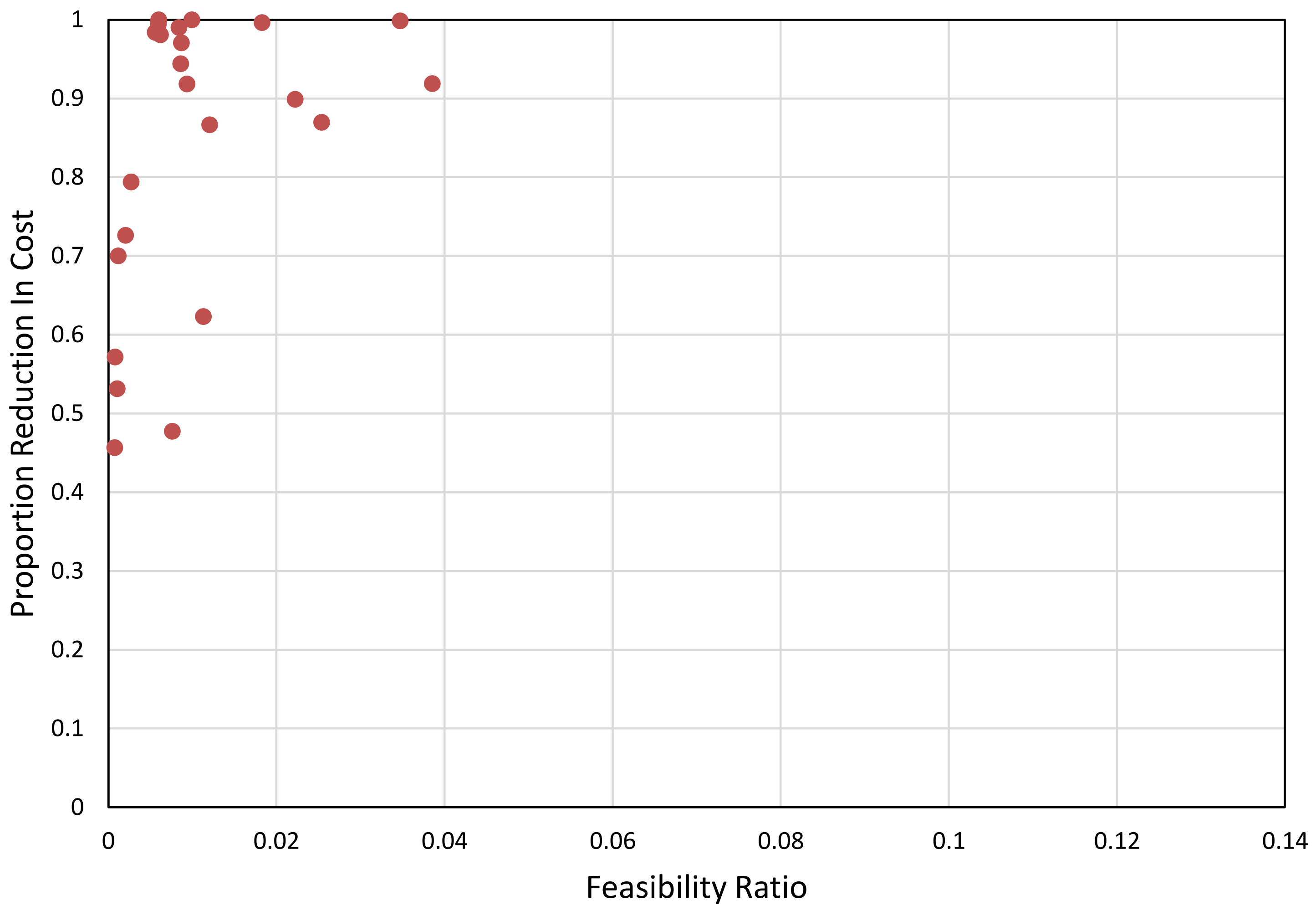
The “Ranking score” is calculated by comparing against all other finalists in the competition



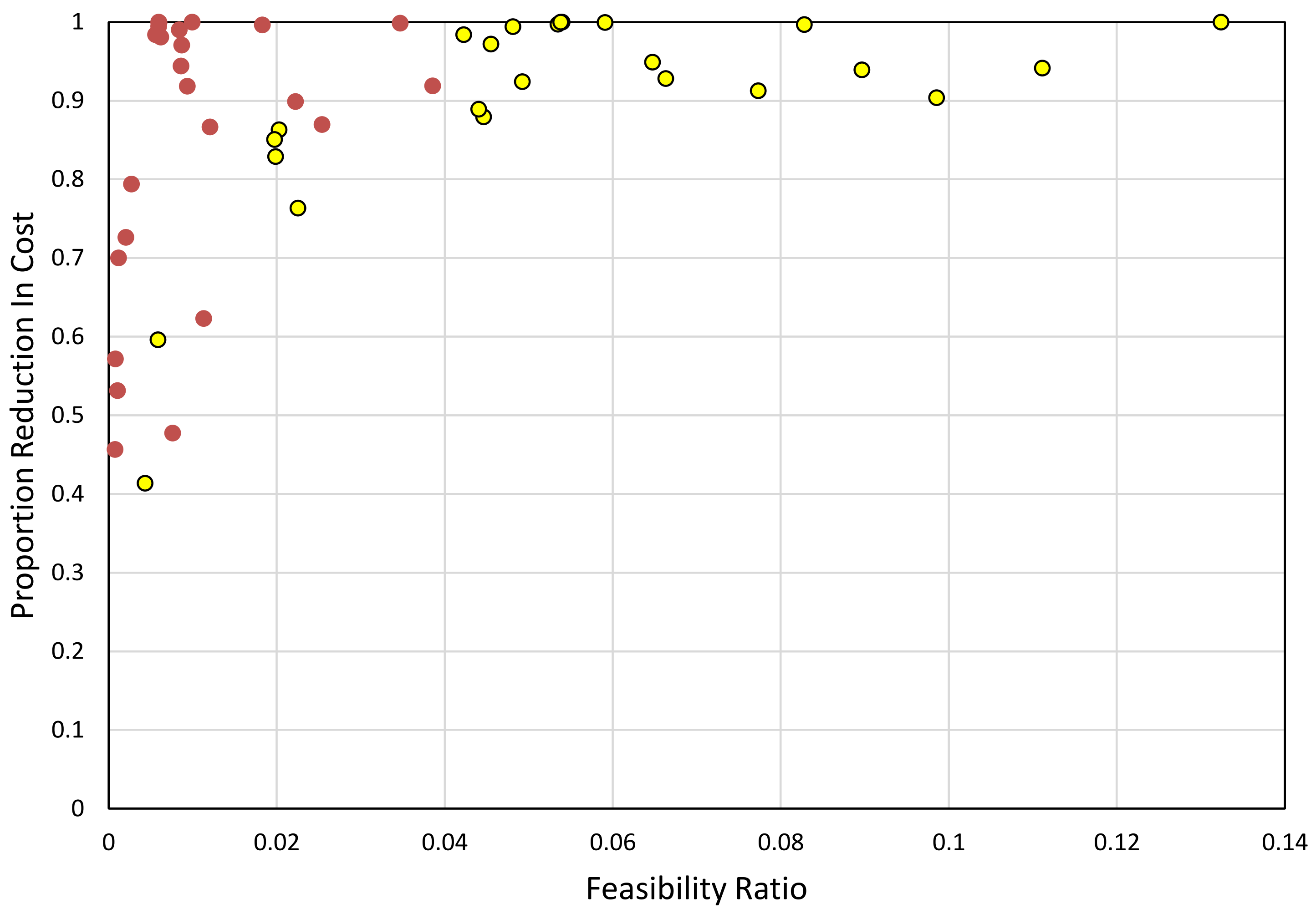
Feasibility Ratio Vs Reduction in Cost using N_1 on the 24 available problem instances.



Feasibility Ratio Vs Reduction in Cost using N_1 and N_2 on the 24 available problem instances.



Feasibility Ratio Vs Reduction in Cost using N_2 on the 24 available problem instances.

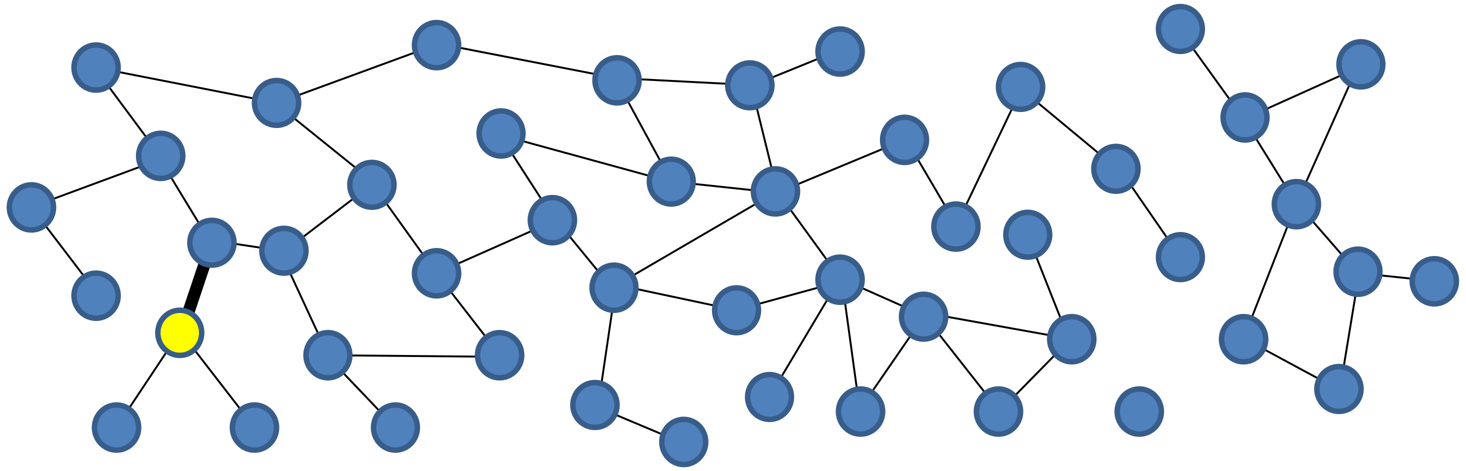


Feasibility Ratio Vs Reduction in Cost using N_2 and N_3 on the 24 available problem instances.

Simulated Annealing and Tabu Search

- In general, for this problem SA seems better at finding good solutions in the space of feasible solutions.
- Perhaps this is because tabu search eliminates additional edges, further reducing connectivity…

- Tabu Search…

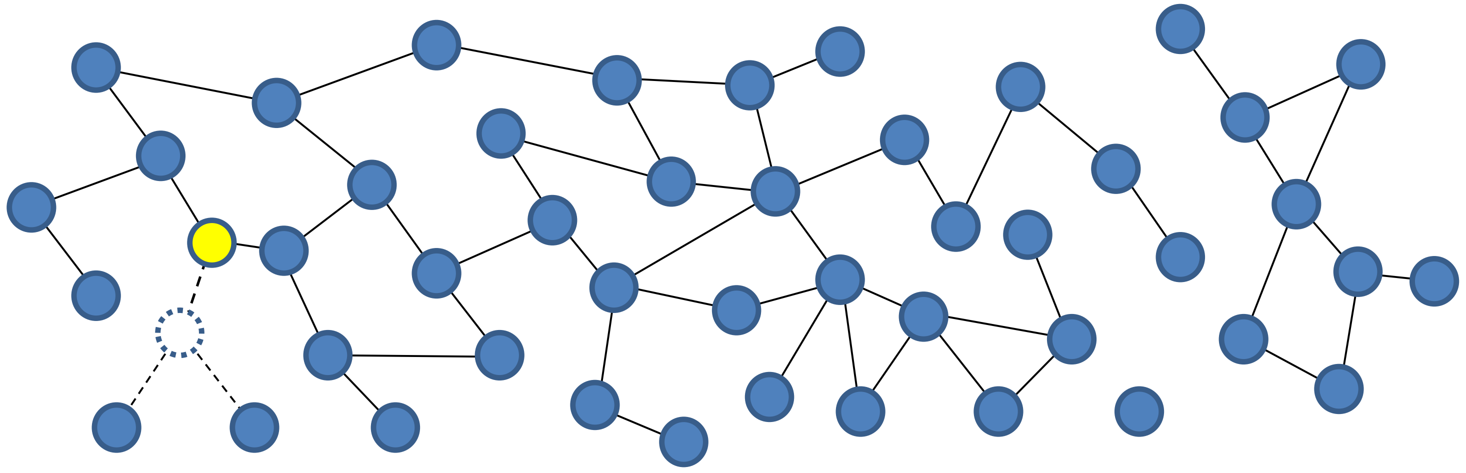


Space of all feasible solutions. Edges indicate the existence of a neighbourhood move from one solution to another

Simulated Annealing and Tabu Search

- In general, for this problem SA seems better at finding good solutions in the space of feasible solutions.
- Perhaps this is because tabu search eliminates additional edges, further reducing connectivity…

- Tabu Search…

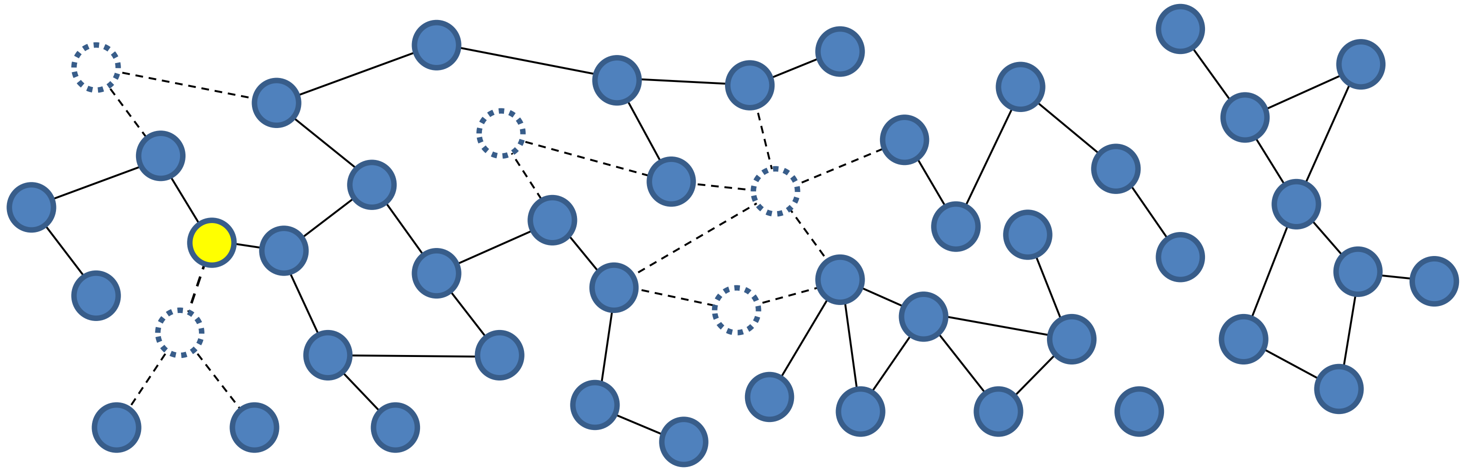


Space of all feasible solutions. Edges indicate the existence of a neighbourhood move from one solution to another

Simulated Annealing and Tabu Search

- In general, for this problem SA seems better at finding good solutions in the space of feasible solutions.
- Perhaps this is because tabu search eliminates additional edges, further reducing connectivity…

- Tabu Search…

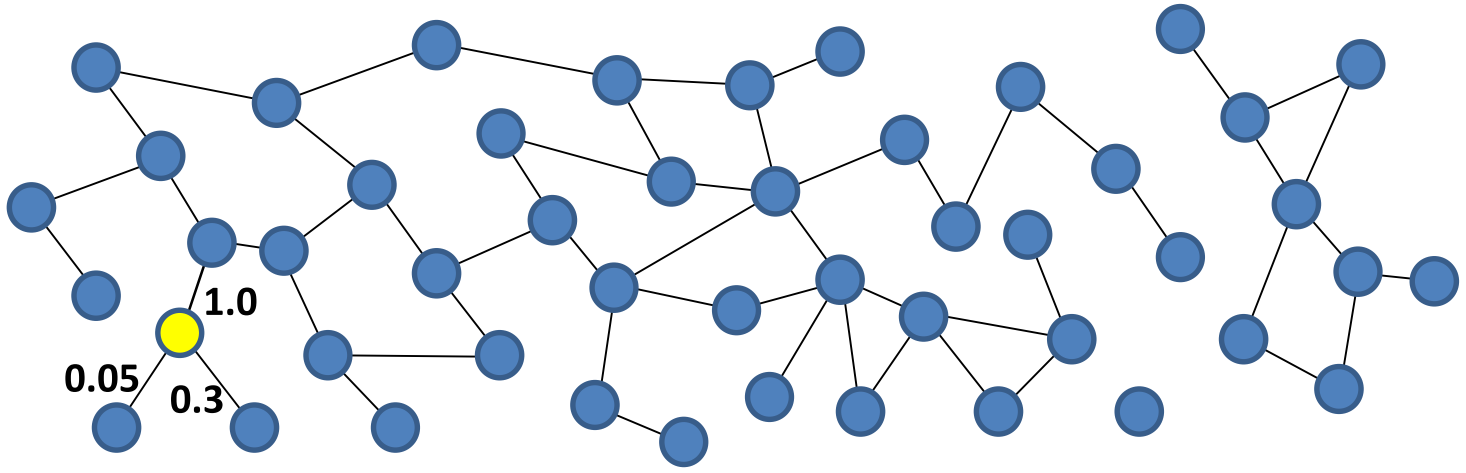


Space of all feasible solutions. Edges indicate the existence of a neighbourhood move from one solution to another

Simulated Annealing and Tabu Search

- In general, for this problem SA seems better at finding good solutions in the space of feasible solutions.
- Perhaps this is because tabu search eliminates additional edges, further reducing connectivity…

- Tabu Search…



Space of all feasible solutions. Edges indicate the existence of a neighbourhood move from one solution to another

Simulated Annealing: One-Stage Vs Two-Stage

- An alternative approach is to consider the larger space of feasible **and** infeasible solutions, and then apply SA using a weighted cost function
- However, do one-stage approaches benefit from the existence of a zero-cost solution?
- In other words, by moving towards solutions with few soft constraint violations, do they also happen to move towards feasible regions of the search space too?

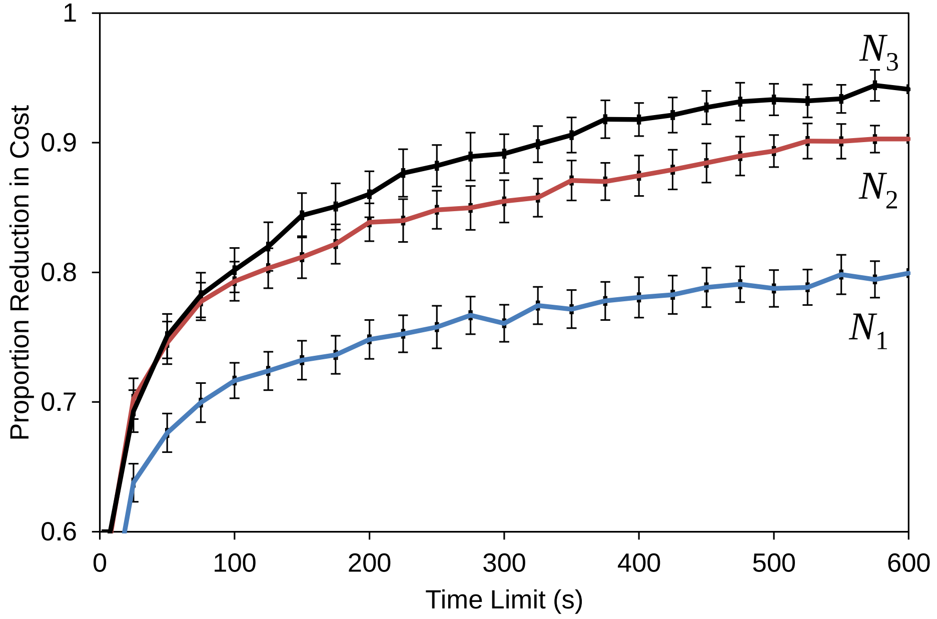
	Perfect Solution Known to Exist	Perfect Solution not Known to Exist
Two Stage SA (this method)	4.5	21
One Stage SA with weighted cost function*	12.5	2

Number of instances (out of 40) where each method outperforms the other.

*Ceschia, et al. (2012) "Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem". Computers and Operational Research, 39:1615–1624

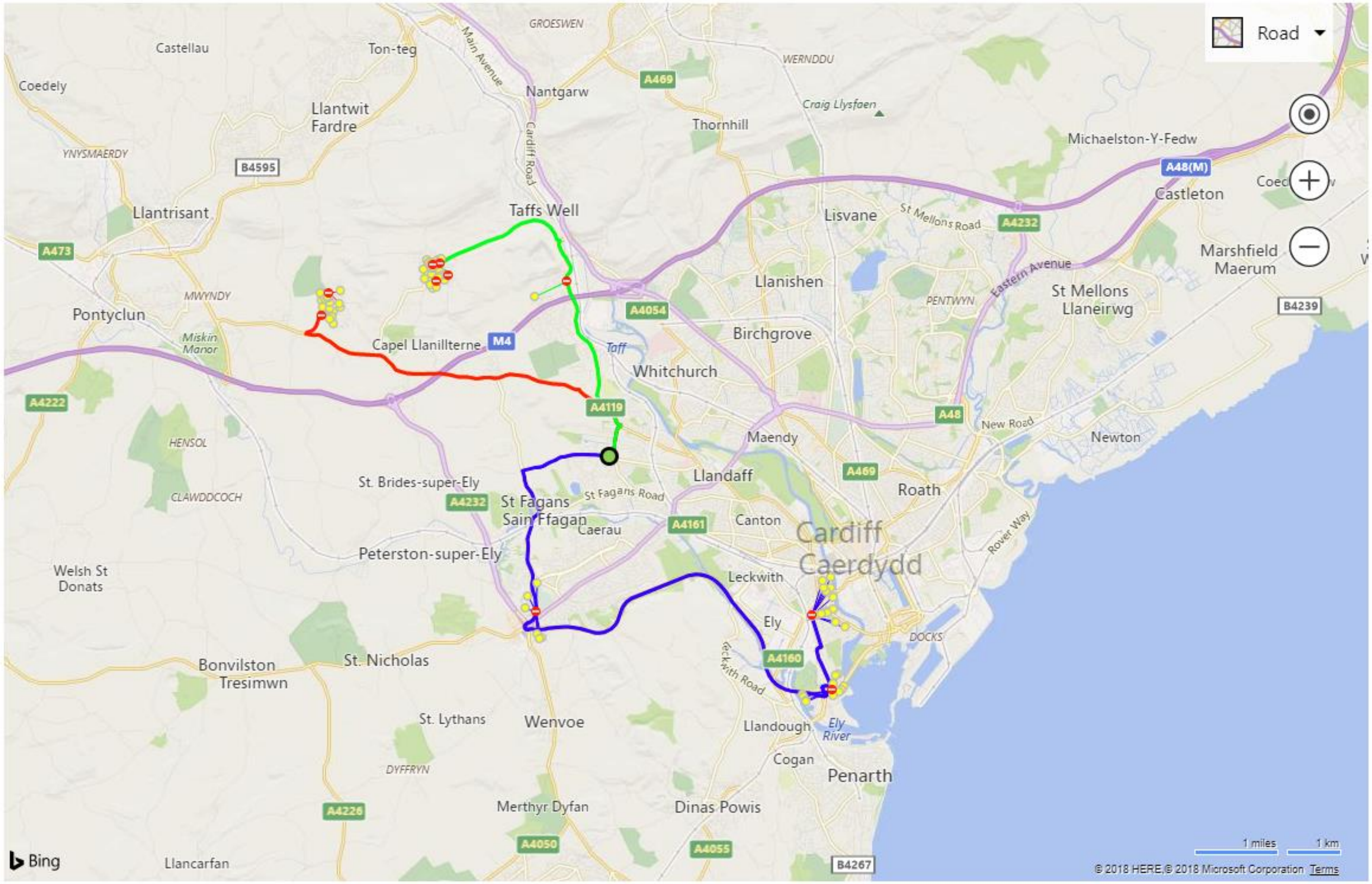
Summary on the Timetabling Problem

- Better results can be achieved for this problem if we improve the feasibility ratio.
- Suitable neighborhood operators can be formulated by making note of the underlying structures of the problem.
- SA seems to be a successful methodology for this problem. In our method the only parameters needed are
  - End temperature
  - **Time Limit**



<u>Proportion decrease in cost using differing time limits and differing neighbourhood operators.</u>

Problem II: School Bus Transport



School Bus Transport

The Current Procedure…

1) Organised by local government

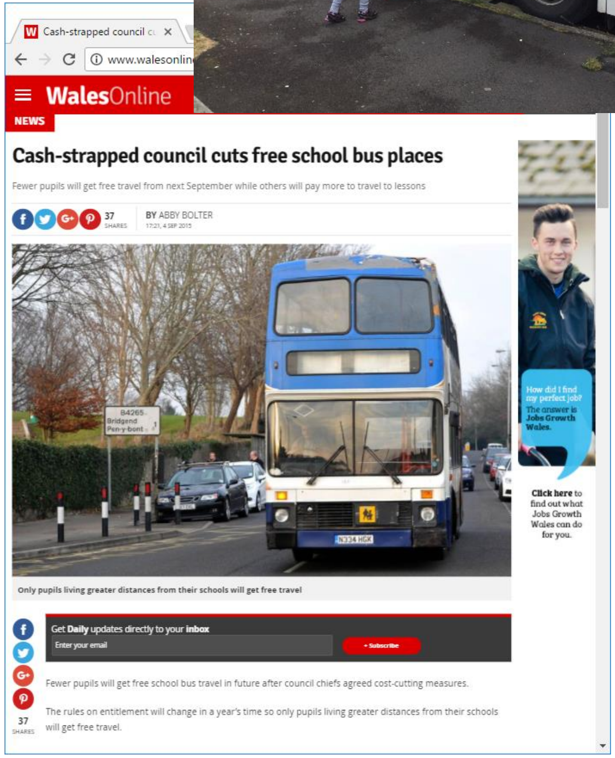
2) For each school a list of **eligible** addresses is compiled

3) A set of suitable bus routes are created to serve all qualifying students.

4) Bus companies then bid for the contracts.

- Yearly contract for a 70-seat bus typically GBP£25,000 to £35,000,

- Costs can increase for longer journeys and for routes requiring a chaperone



Cash-strapped council cu

www.walesonlin

WalesOnline

NEWS

Cash-strapped council cuts free school bus places

Fewer pupils will get free travel from next September while others will pay more to travel to lessons

37 SHARES

BY ABBY BOLTER
17:21, 4 SEP 2015

Only pupils living greater distances from their schools will get free travel

Get **Daily** updates directly to your **inbox**

Enter your email

+ Subscribe

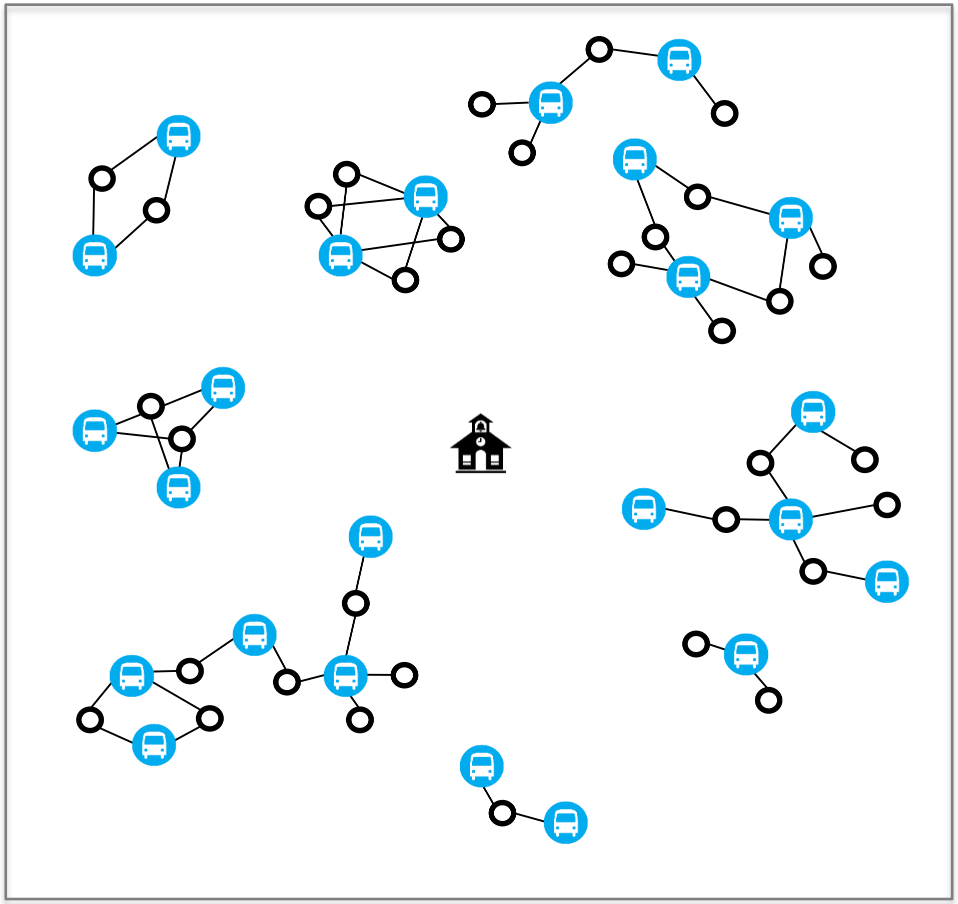
Fewer pupils will get free school bus travel in future after council chiefs agreed cost-cutting measures.

The rules on entitlement will change in a year's time so only pupils living greater distances from their schools will get free travel.

How did I find my perfect job? The answer is Jobs Growth Wales.

Click here to find out what Jobs Growth Wales can do for you.

Problem Description



● Bus Stop
○ Address

Constraints...

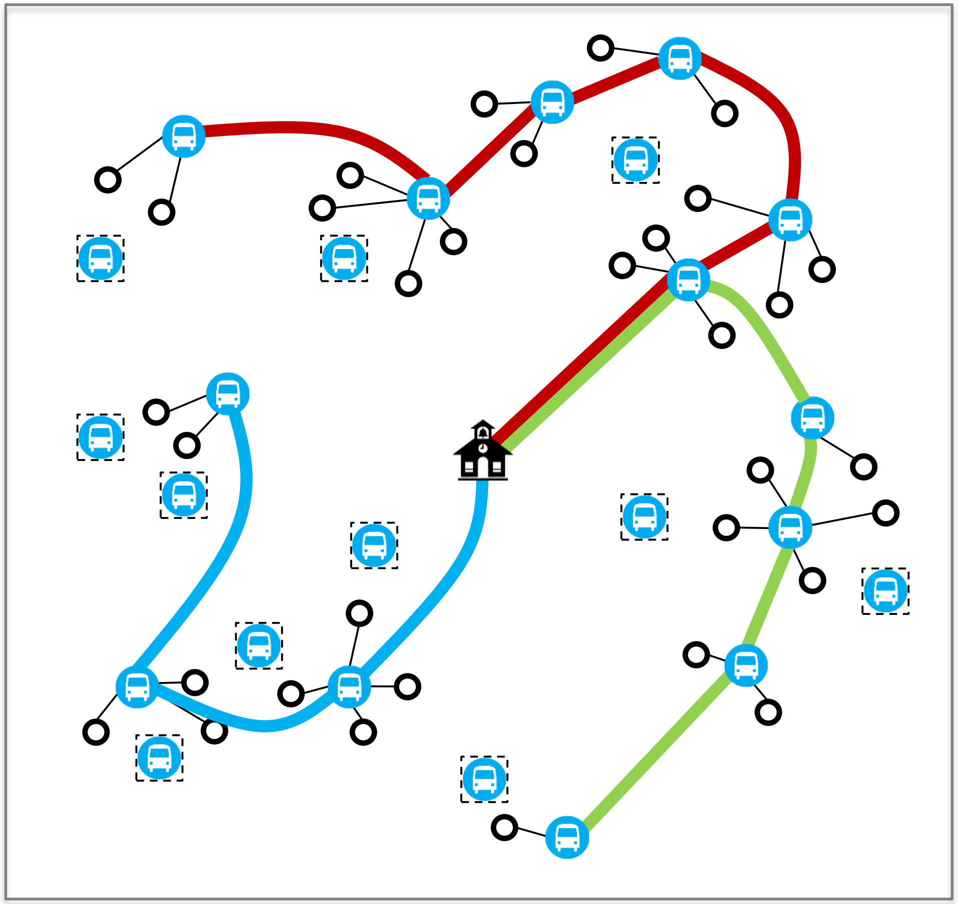
1) Bus journeys should not be too long (<45 mins)
2) Stops should within walking distance from home (<1 mile)

Features...

1) Minimise the number of buses / routes
2) Use a subset of stops
3) Multi-stops are permitted
4) Boarding the bus takes time

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

Problem Description



🚌	Bus Stop
○	Address

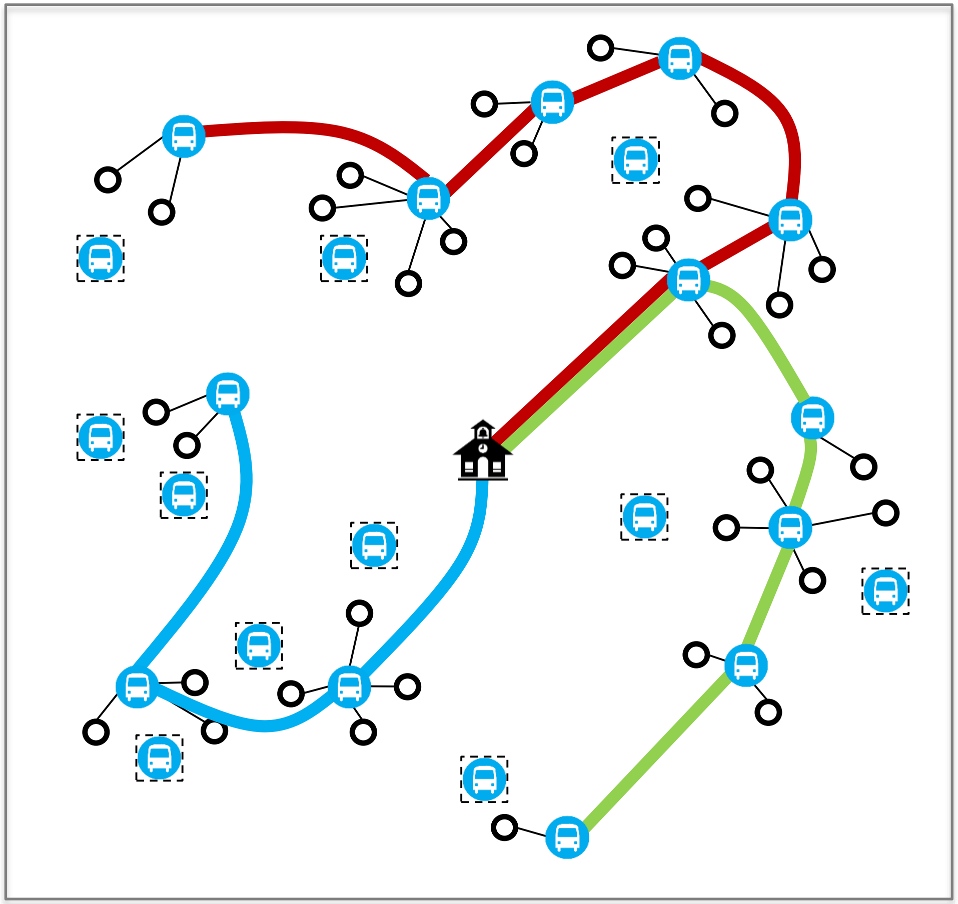
Constraints…

1) Bus journeys should not be too long (<45 mins)
2) Stops should within walking distance from home (<1 mile)

Features…

1) Minimise the number of buses / routes
2) Use a subset of stops
3) Multi-stops are permitted
4) Boarding the bus takes time

Problem Description



🔵	Bus Stop
○	Address

A Feasible Solution...

1) All addresses must have a serviced bus stop within walking distance

2) Journeys do not exceed the maximum time limit

3) Number of students boarding does not exceed maximum bus capacity

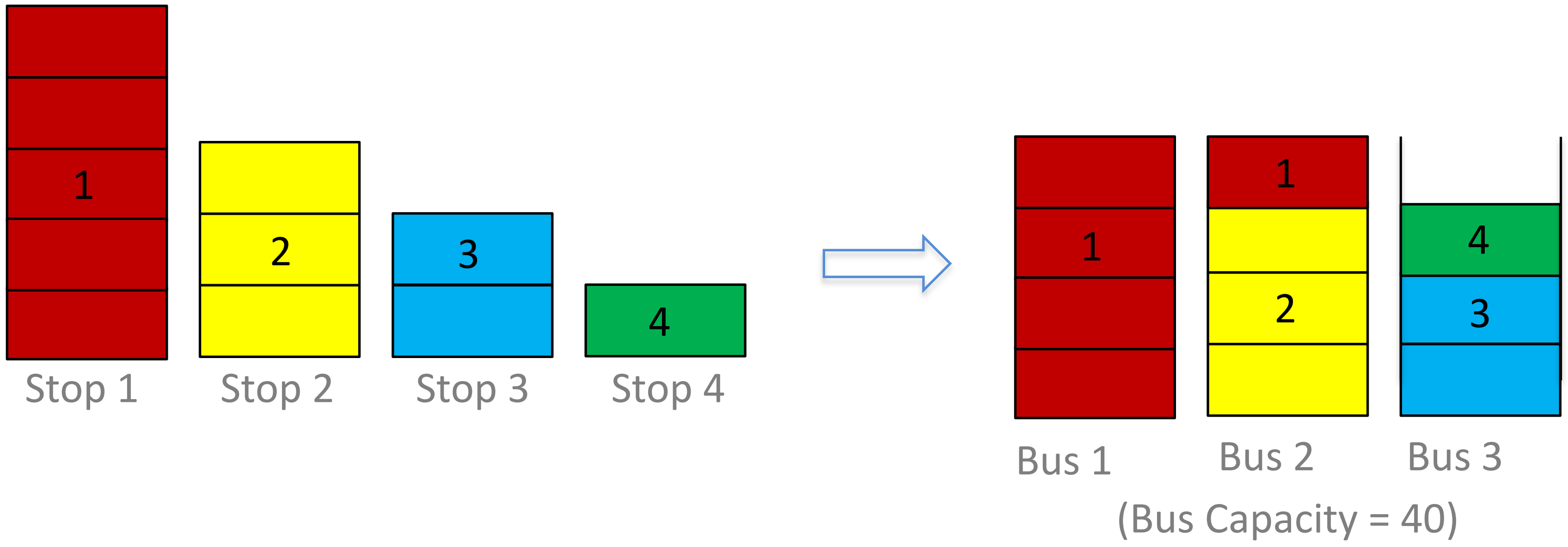
Allocating students to routes/buses

- Stop 1, 50 students
- Stop 2, 30 students
- Stop 3, 20 students
- Stop 4, 10 students

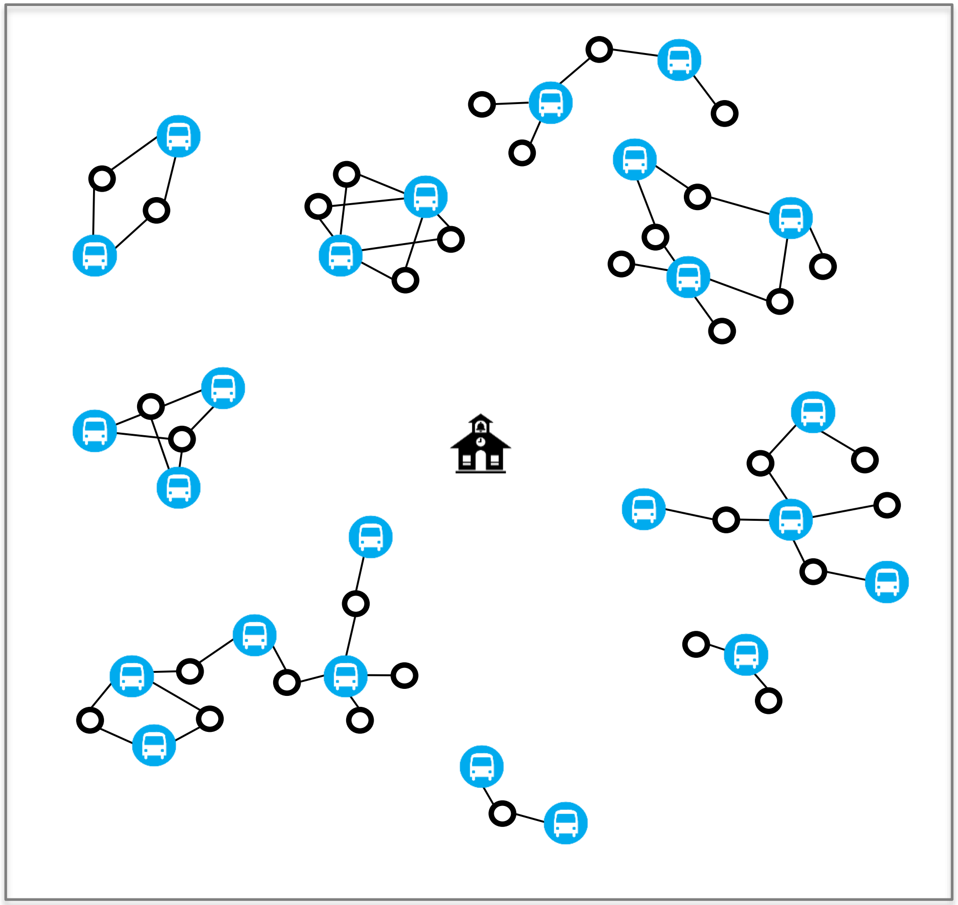
Students are allocated to the closest stops being used

This results in a relaxed **bin packing problem.**

Splitting an "item" results in a multi-stop



Selecting the subset of stops



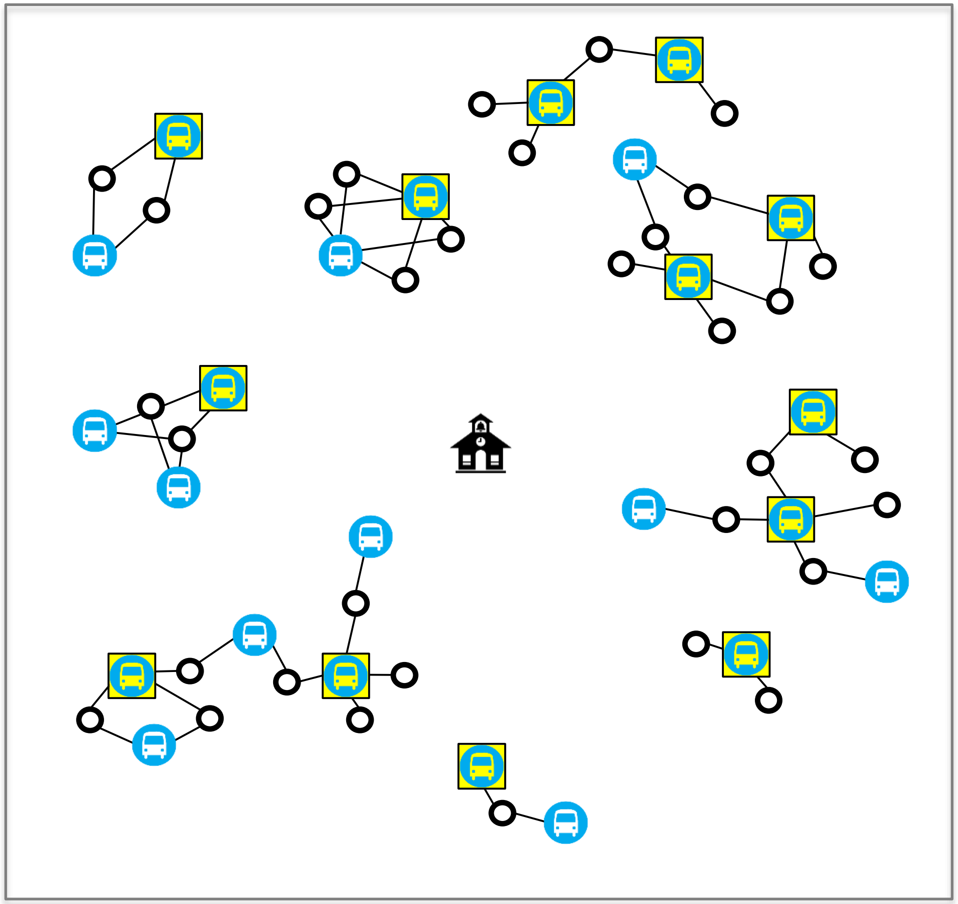
● Bus Stop
○ Address

Let **S** be the set whose elements correspond to the addresses within walking distance of each bus stop:

All addresses in a feasible solution must be served by a bus stop;

Hence the task of choosing a suitable subset of stops is a **set covering problem** using **S** and the set of stops.

Selecting the subset of stops



Let **S** be the set whose elements correspond to the addresses within walking distance of each bus stop:

All addresses in a feasible solution must be served by a bus stop;

Hence the task of choosing a suitable subset of stops is a **set covering problem** using **S** and the set of stops.

Theorem

Assuming

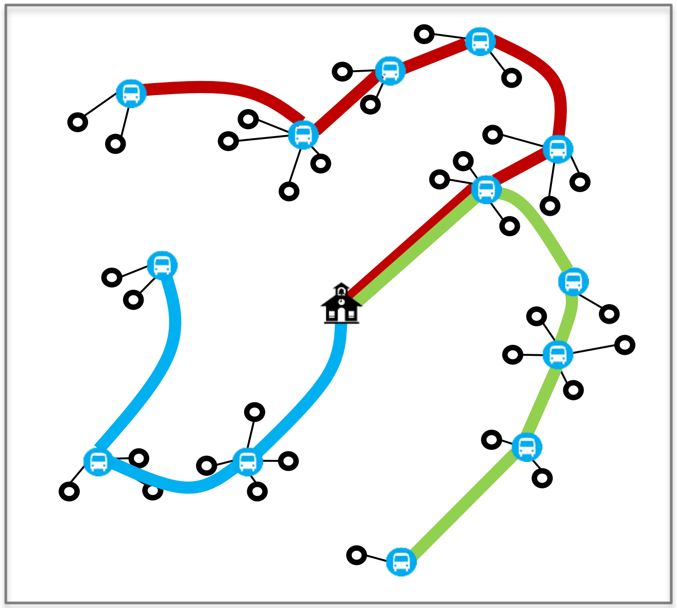
- the triangle inequality, and
- multistops are not permitted,

the optimal solution corresponds to a **minimal** set covering

Algorithm Strategy

Only consider feasible solutions, but allow long routes.

Then seek to shorten the routes to below the required time limit



For a fixed number of vehicles k…

1) Create a minimal covering of stops, assign all passengers to stops, and all stops to vehicles.

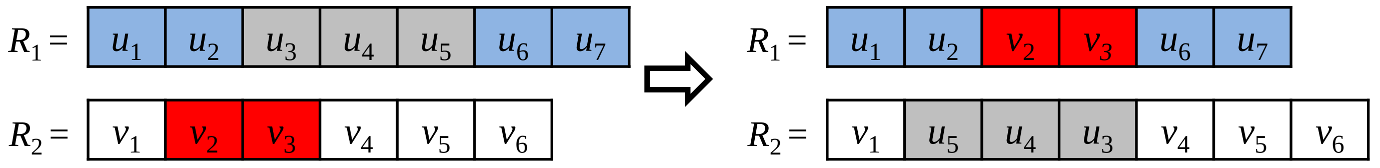
2) Use a local search operator to shorten the resultant routes.

3) Use the current solution to determine a new minimal covering of stops and repair the solution.

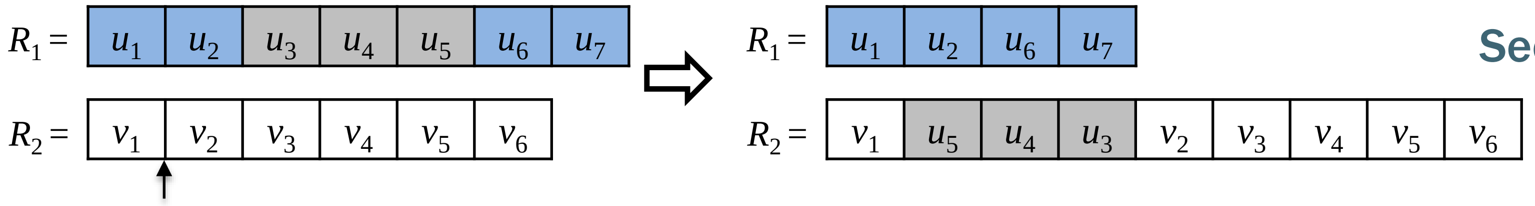
4) Return to 2) **OR** increase k and return to 1)

Local Search Operators

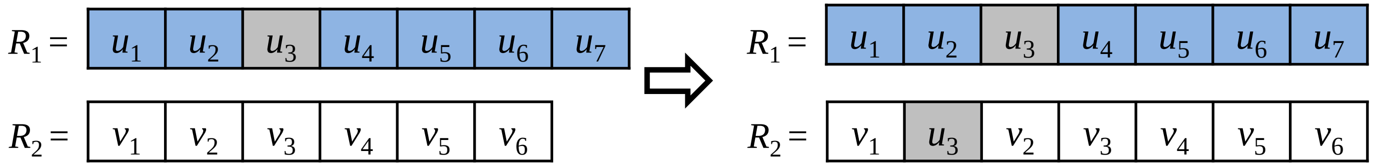
Inter-route Operators


$R_1 = u_1\ u_2\ u_3\ u_4\ u_5\ u_6\ u_7$
$R_2 = v_1\ v_2\ v_3\ v_4\ v_5\ v_6$
$R_1 = u_1\ u_2\ v_2\ v_3\ u_6\ u_7$
$R_2 = v_1\ u_5\ u_4\ u_3\ v_4\ v_5\ v_6$

Section Swap

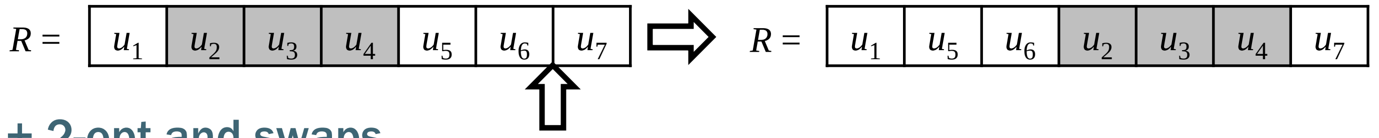

$R_1 = u_1\ u_2\ u_3\ u_4\ u_5\ u_6\ u_7$
$R_2 = v_1\ v_2\ v_3\ v_4\ v_5\ v_6$
$R_1 = u_1\ u_2\ u_6\ u_7$
$R_2 = v_1\ u_5\ u_4\ u_3\ v_2\ v_3\ v_4\ v_5\ v_6$

Section Insert


$R_1 = u_1\ u_2\ u_3\ u_4\ u_5\ u_6\ u_7$
$R_2 = v_1\ v_2\ v_3\ v_4\ v_5\ v_6$
$R_1 = u_1\ u_2\ u_3\ u_4\ u_5\ u_6\ u_7$
$R_2 = v_1\ u_3\ v_2\ v_3\ v_4\ v_5\ v_6$

Create Multi-stop

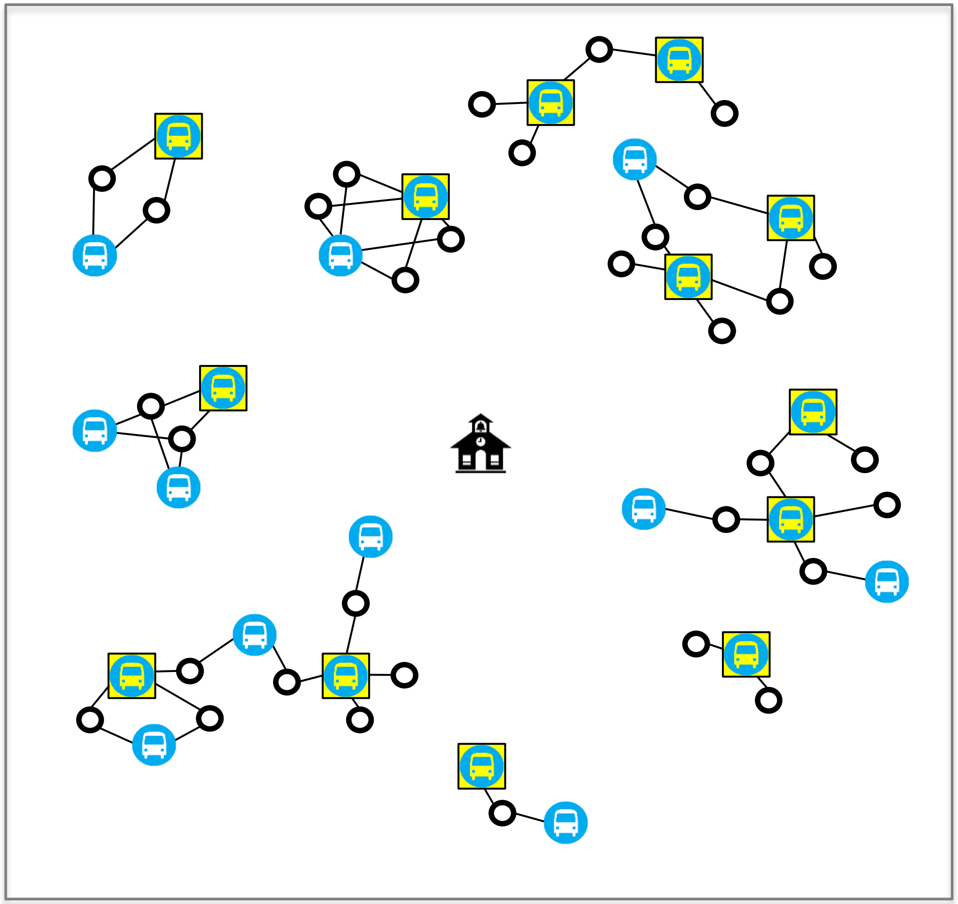
Intra-route Operators


$R = u_1\ u_2\ u_3\ u_4\ u_5\ u_6\ u_7$
$R = u_1\ u_5\ u_6\ u_2\ u_3\ u_4\ u_7$

Extended Or-Opt

+ 2-opt and swaps

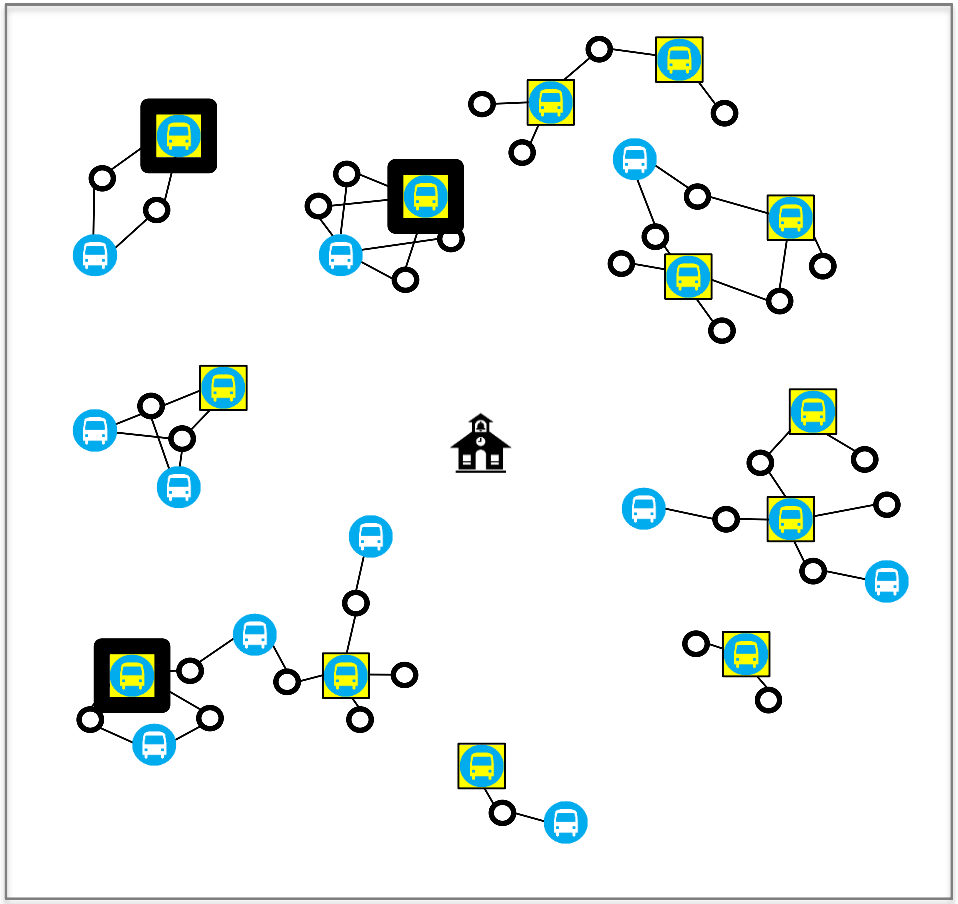
Generating a new minimal subset of stops



●	Bus Stop
○	Address

1. **Take a small number of non-compulsory serviced stops and deselect them.**
2. Add new stops to ensure all passengers are served. (Ensuring the subset is still minimal).
3. Repair the routes and passenger allocations to reflect the changes

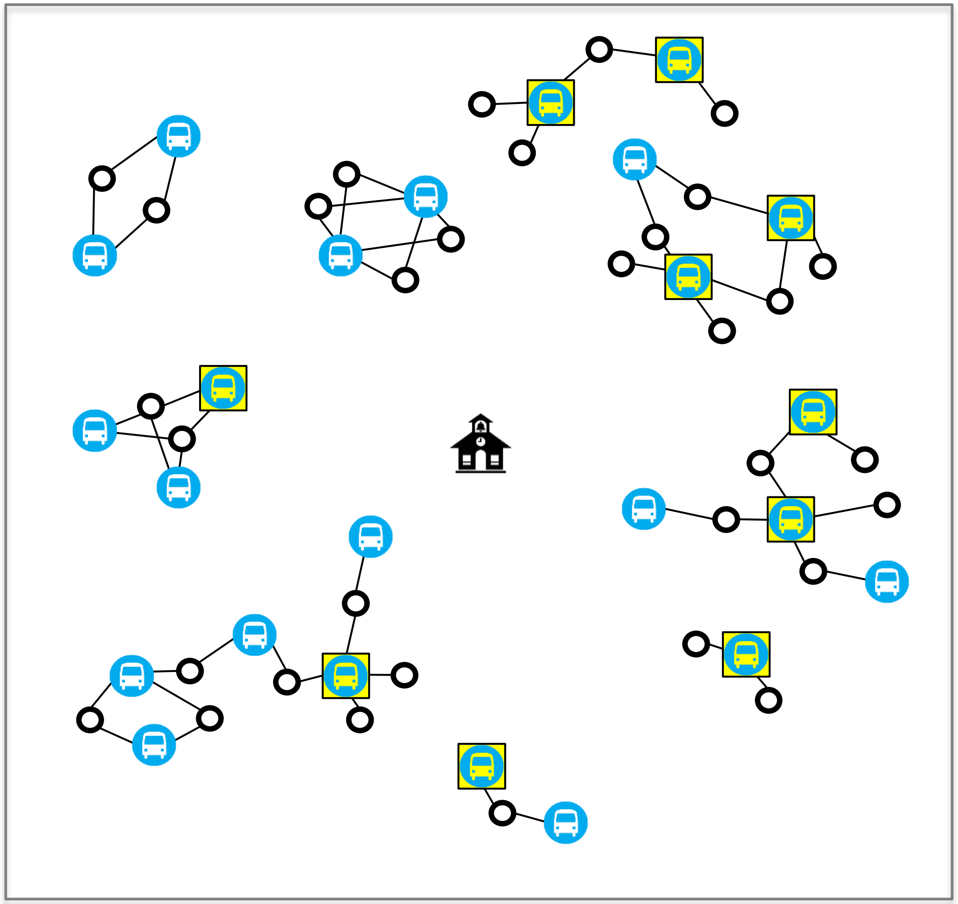
Generating a new minimal subset of stops



🚌 Bus Stop
○ Address

1. **Take a small number of non-compulsory serviced stops and deselect them.**
2. Add new stops to ensure all passengers are served. (Ensuring the subset is still minimal).
3. Repair the routes and passenger allocations to reflect the changes

Generating a new minimal subset of stops

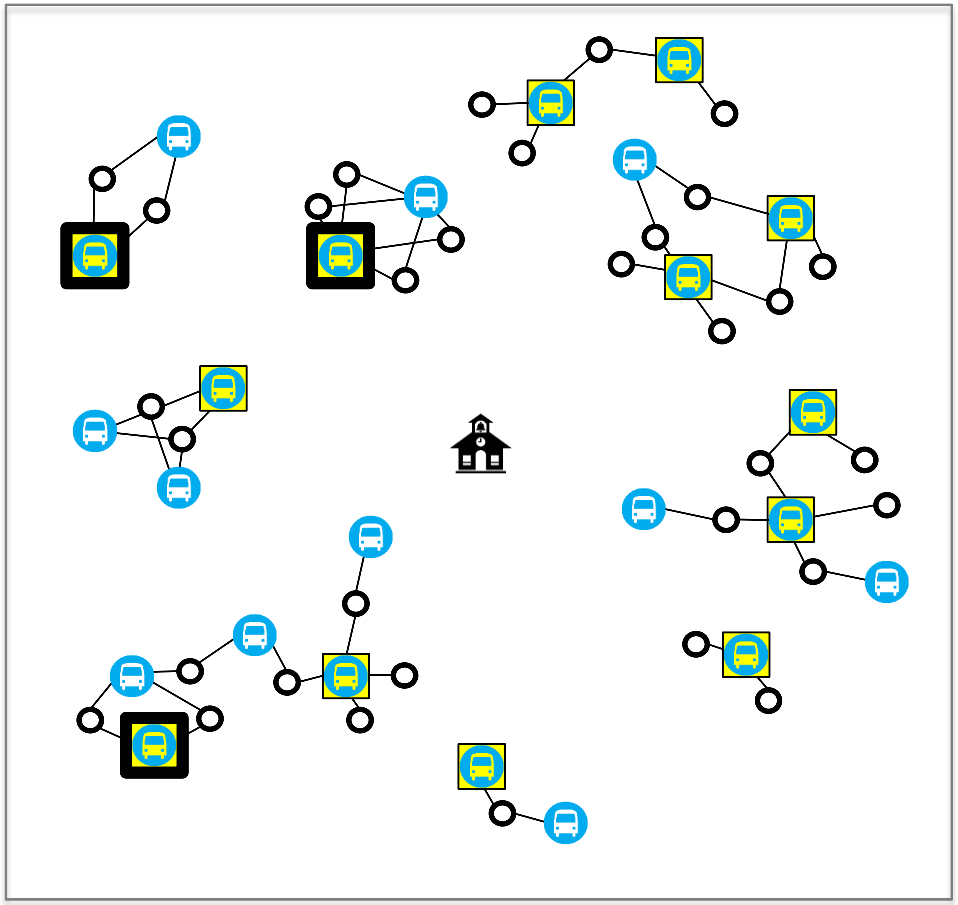


Bus Stop

Address

1. **Take a small number of non-compulsory serviced stops and deselect them.**
2. Add new stops to ensure all passengers are served. (Ensuring the subset is still minimal).
3. Repair the routes and passenger allocations to reflect the changes

Generating a new minimal subset of stops



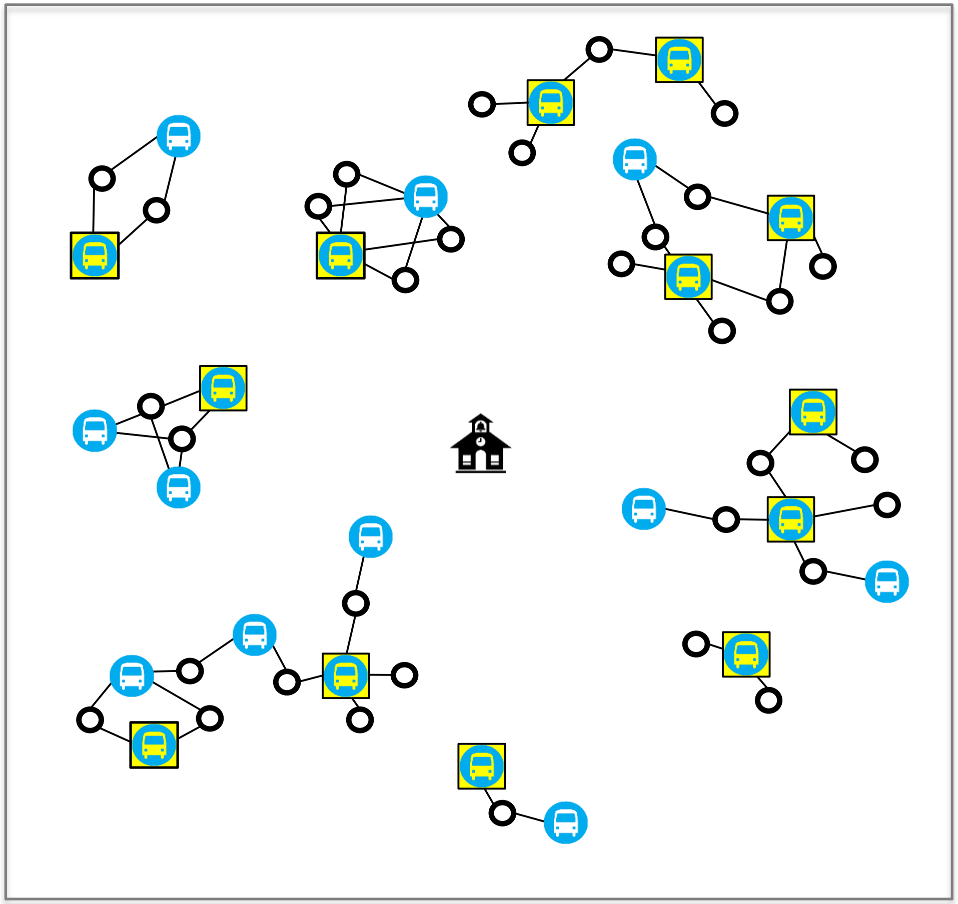
Bus Stop
Address

1. Take a small number of non-compulsory serviced stops and deselect them.

2. **Add new stops to ensure all passengers are served. (Ensuring the subset is still minimal).**

3. Repair the routes and passenger allocations to reflect the changes

Generating a new minimal subset of stops

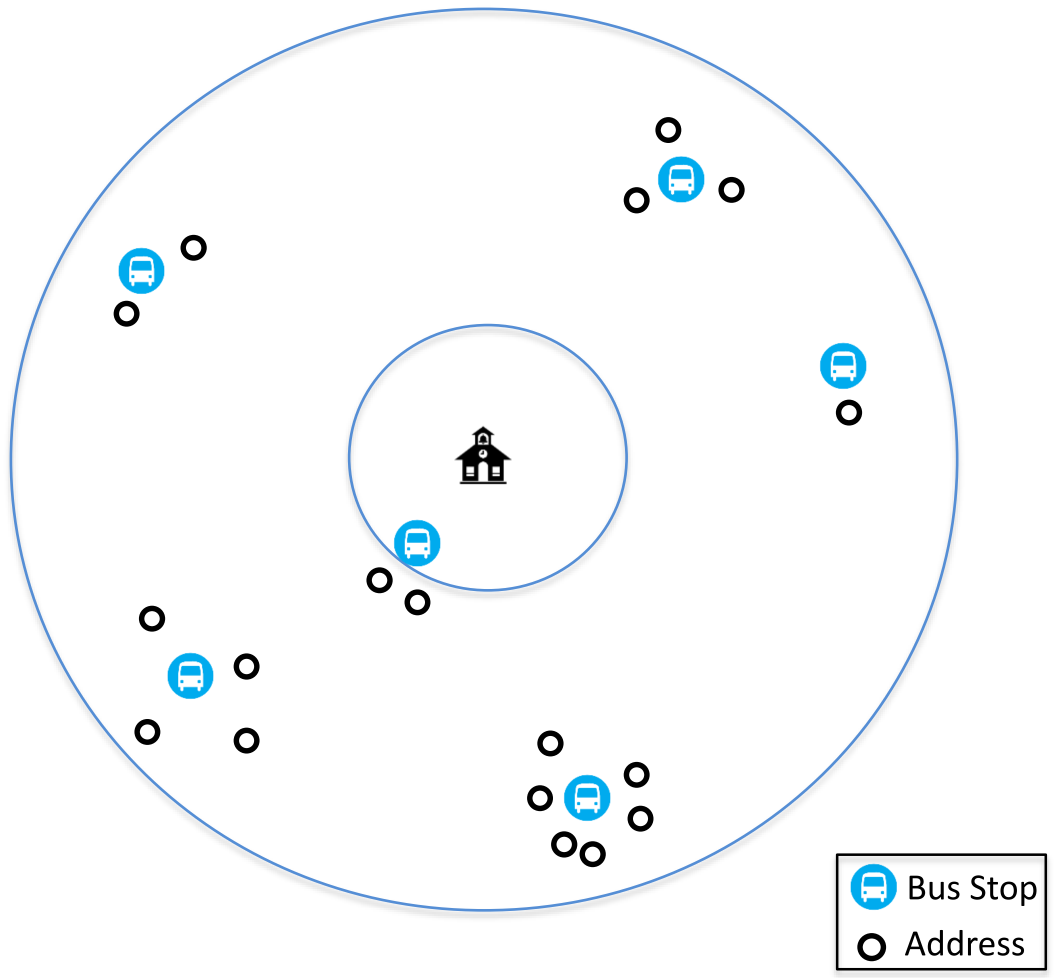


1. Take a small number of non-compulsory serviced stops and deselect them.

2. **Add new stops to ensure all passengers are served. (Ensuring the subset is still minimal).**

3. Repair the routes and passenger allocations to reflect the changes

Random Instance Generation

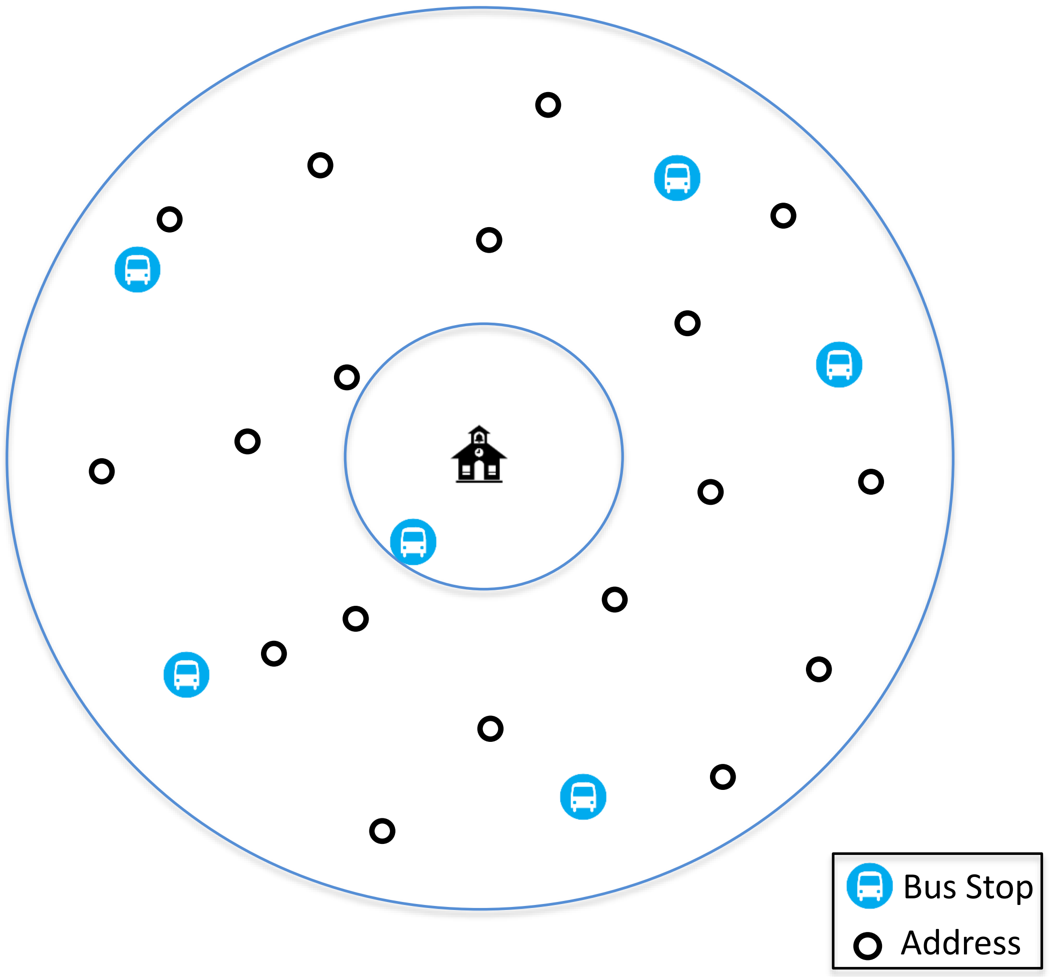


Put a school at the centre of a circle…

1) Add stops anywhere in the circle
2) Now add addresses that are
   A. Within walking distance of a stop
   B. Not too close to the school.
3) Finally, remove any stops with no address within waking distance

CARDIFF UNIVERSITY PRIFYSGOL CAERDYDD

Random Instance Generation



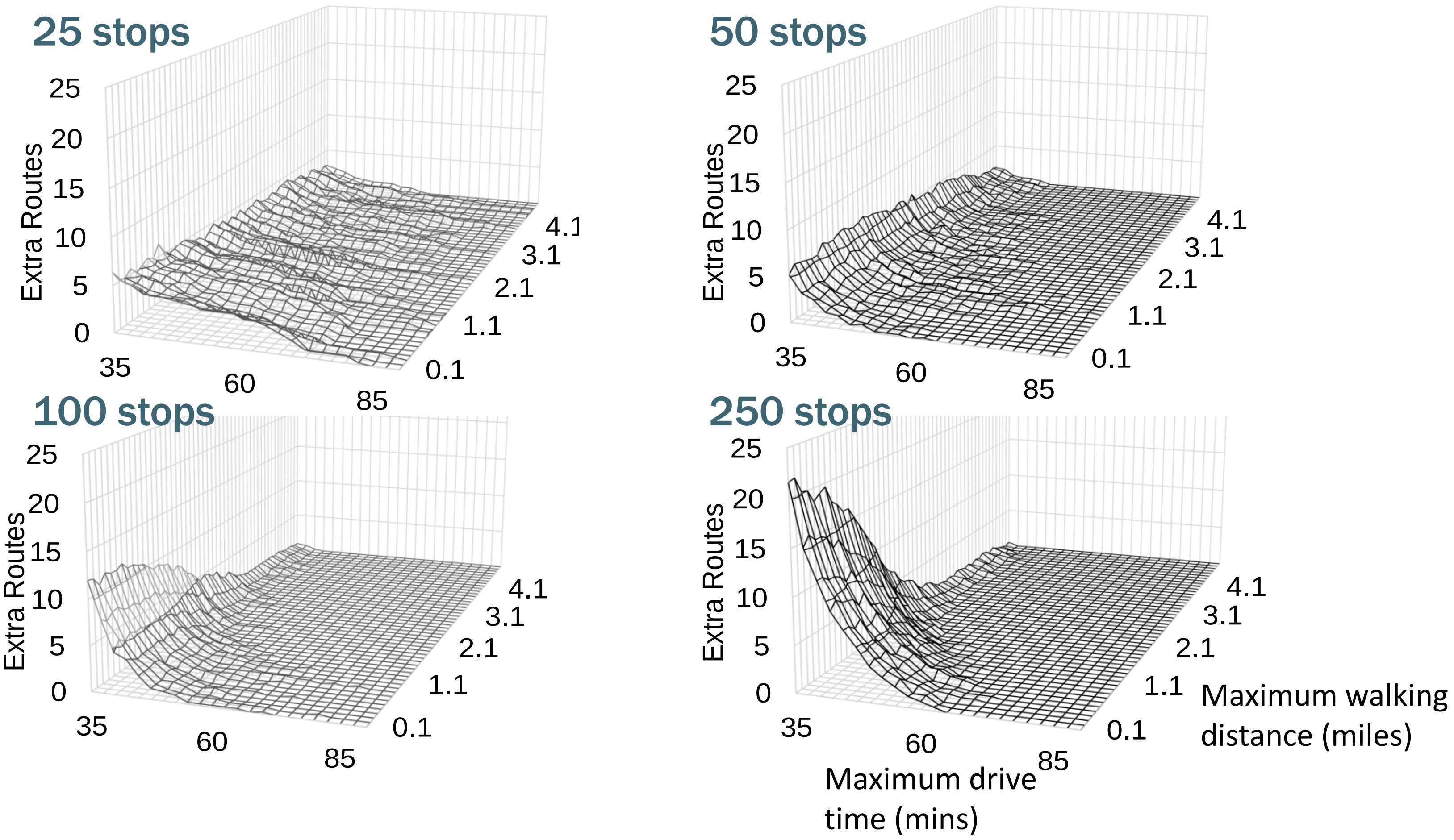
Put a school at the centre of a circle…

1) Add stops anywhere in the circle

2) Now add addresses that are
   A. Within walking distance of a stop
   B. Not too close to the school.

3) Finally, remove any stops with no address within waking distance

Results with Random Graphs



Extra vehicles (routes) required for random graphs with 1,000 students using 70-seat buses. All instances used a 15-mile radius circle, with buses travelling along straight lines at 30mph; hence, all bus stops are within 30 minutes of the school.

Summary and Discussion

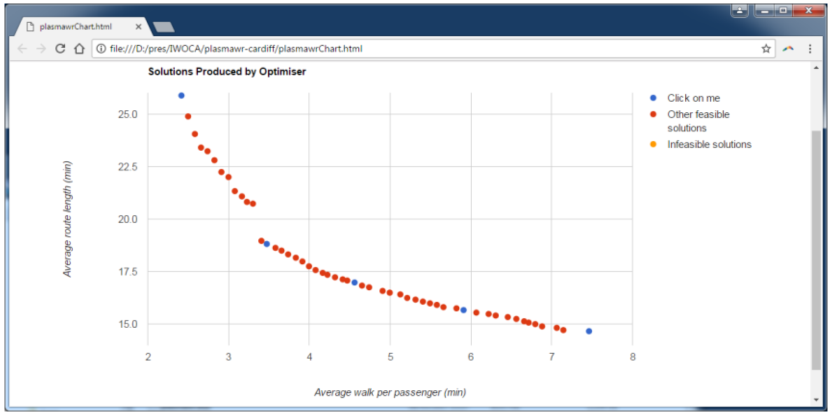
- The lower bound on the number of vehicles

$$\text{lower bound} = \left\lceil \frac{\text{number of students}}{\text{bus capacity}} \right\rceil$$

is usually achieved quickly (for random and real-world instances)

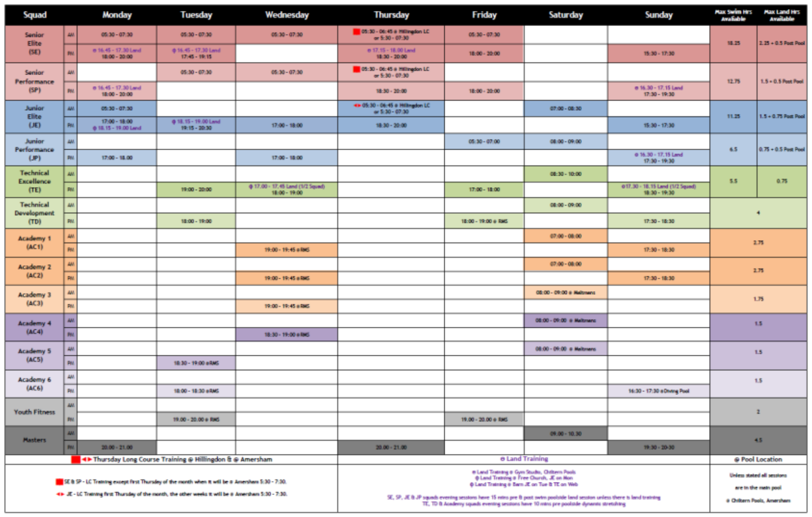
- In real-world problems, use of minimal coverings seems to result in overly long walks

- We must also consider the multi-objective nature of the problem



www.rhydlewis.eu/bus

Two Example Optimisation Problems from the World of Education



Squad		Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday	Max Swim Hrs Available	Max Land Hrs Available
Senior Elite (SE)	AM	05:30 - 07:30	05:30 - 07:30	05:30 - 07:30	🟥 05:30 - 06:45 @ Hillingdon LC or 5:30 - 07:30	05:30 - 07:30			18.25	2.25 + 0.5 Post Pool
	PM	⊕ 16.45 - 17.30 Land 18:00 - 20:00	ɸ 16.45 - 17.30 Land 17:45 - 19:15		⊕ 17.15 - 18.00 Land 18:30 - 20:00	18:00 - 20:00		15:30 - 17:30		
Senior Performance (SP)	AM		05:30 - 07:30	05:30 - 07:30	🟥 05:30 - 06:45 @ Hillingdon LC or 5:30 - 07:30				12.75	1.5 + 0.5 Post Pool
	PM	⊕ 16.45 - 17.30 Land 18:00 - 20:00			18:30 - 20:00	18:00 - 20:00		⊕ 16.30 - 17.15 Land 17:30 - 19:30		
Junior Elite (JE)	AM	05:30 - 07:30			◂▸ 05:30 - 06:45 @ Hillingdon LC or 5:30 - 07:30		07:00 - 08:30		11.25	1.5 + 0.75 Post Pool
	PM	17:00 - 18:00 ɸ 18.15 - 19.00 Land	ɸ 18.15 - 19.00 Land 19:15 - 20:30	17:00 - 18:00	18:30 - 20:00			15:30 - 17:30		
Junior Performance (JP)	AM					05:30 - 07:00	08:00 - 09:00		6.5	0.75 + 0.5 Post Pool
	PM	17:00 - 18.00		17:00 - 18:00				⊕ 16.30 - 17.15 Land 17:30 - 19:30		
Technical Excellence (TE)	AM						08:30 - 10:00		5.5	0.75
	PM		19:00 - 20:00	ɸ 17.00 - 17.45 Land (1/2 Squad) 18:00 - 19:00		17:00 - 18:00		⊕17.30 - 18.15 Land (1/2 Squad) 18:30 - 19:30		
Technical Development (TD)	AM						08:00 - 09:00		4	
	PM		18:00 - 19:00			18:00 - 19:00 @ RMS		17:30 - 18:30		
Academy 1 (AC1)	AM						07:00 - 08:00		2.75	
	PM			19:00 - 19:45 @RMS				17:30 - 18:30		
Academy 2 (AC2)	AM						07:00 - 08:00		2.75	
	PM			19:00 - 19:45 @RMS				17:30 - 18:30		
Academy 3 (AC3)	AM						08:00 - 09:00 @ Maltmans		1.75	
	PM			19:00 - 19:45 @RMS						
Academy 4 (AC4)	AM						08:00 - 09:00 @ Maltmans		1.5	
	PM			18:30 - 19:00 @RMS						
Academy 5 (AC5)	AM						08:00 - 09:00 @ Maltmans		1.5	
	PM		18:30 - 19:00 @RMS							
Academy 6 (AC6)	AM								1.5	
	PM		18:00 - 18:30 @RMS					16:30 - 17:30 @Diving Pool		
Youth Fitness	AM								2	
	PM		19.00 - 20.00 @ RMS			19.00 - 20.00 @ RMS				
Masters	AM						09.00 - 10.30		4.5	
	PM	20.00 - 21.00			20.00 - 21.00			19:30 - 20:30		

🟥 ◂▸ Thursday Long Course Training @ Hillingdon & @ Amersham

🟥 SE & SP - LC Training except first Thursday of the month when it will be @ Amersham 5:30 - 7:30.

◂▸ JE - LC Training first Thursday of the month, the other weeks it will be @ Amersham 5:30 - 7:30.

⊕ Land Training

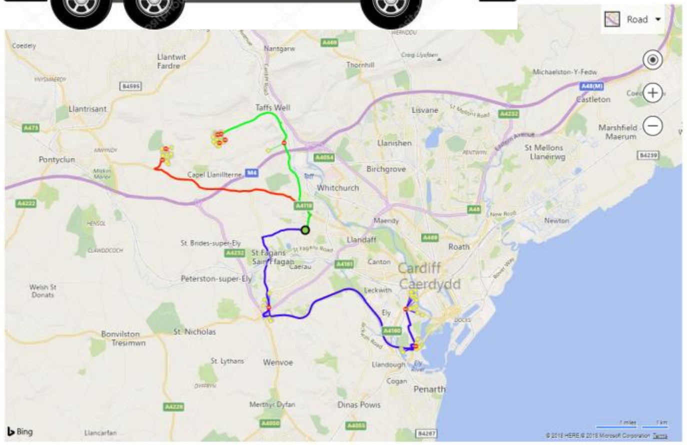
⊕ Land Training @ Gym Studio, Chiltern Pools
ɸ Land Training @ Free Church, JE on Mon
ɸ Land Training @ Barn JE on Tue & TE on Wed

SE, SP, JE & JP squads evening sessions have 15 mins pre & post swim poolside land session unless there is land training
TE, TD & Academy squads evening sessions have 10 mins pre poolside dynamic stretching

@ Pool Location

Unless stated all sessions are in the main pool

@ Chiltern Pools, Amersham



Lewis, R. and J. Thompson (2015) 'Analysing the Effects of Solution Space Connectivity with an Effective Metaheuristic for the Course Timetabling Problem'. *European Journal of Operational Research*, vol. 240, pp. 637-648.

Lewis, R., K. Smith-Miles, and K. Phillips (2018) 'The School Bus Routing Problem: An Analysis and Algorithm'. In *Combinatorial Algorithms* (LNCS 10765), Springer, pp. 287-298.

Rhyd Lewis
School of Mathematics, Cardiff University,
LewisR9@cf.ac.uk, www.RhydLewis.eu