Acceleration strategies and column generation heuristics

Guy Desaulniers
École Polytechnique of Montréal

School on Column Generation

Paris, March 10-14, 2014

Acceleration strategies

- Start by implementing a basic version of a branch-and-price algorithm

- Run tests to identify weaknesses
  - Number of CG iterations
  - Time in MP versus time in pricing
  - Number of BB nodes

Large number of CG iterations

- Generate multiple columns per iteration
  - Up to 500 or even a little bit more
    - Easy if labeling algorithm or using a local search heuristic
    - If MIP pricing problem, keep all negative reduced cost solutions encountered or solve several times with variables removed
- Use less heuristic pricing algorithms to find columns with large negative reduced costs
- Dual variable stabilization
- Early branching using the DW bound

Most of the time spent in MP

- Degeneracy
  - Dual simplex or barrier algorithm
  - Perturbation
  - Dual variable stabilization
    - Draw dual values over the CG iterations
  - Dynamic constraint aggregation
  - Improved primal simplex (IPS)
- High matrix density
  - Reformulate some constraints

Most of the time spent in MP

- Large number of constraints
  - Relax some constraints and reintroduce them only if needed
- Large number of columns
  - Remove columns occasionally (not at each iteration)
    - They can always be generated again
    - For instances with more than 1000 constraints, the number of columns to keep should be between 2 and 3 times the number of constraints

Most of the time spent in pricing

- Use partial pricing to try to generate negative reduced cost columns
  - Fast heuristics
    - Same as exact algorithm but with less variables
    - Same as exact algorithm but with heuristic rules
    - Local search
  - Do not solve all pricing problems when there are many
    - Solve pricing problems until n (=3) have produced negative reduced cost columns

Most of the time spent in pricing

- Use relaxed pricing problems
  - May yield weaker lower bounds
- Aggregate identical pricing problems
- If nearly identical, compute the reduced cost of generated columns for all the other pricing problems and skip those for which negative reduced cost columns are found in this way

Large number of BB nodes

- Can you strengthen the lower bounds?
  - Stronger pricing problems
    - With integrality requirements
    - Less relaxed
  - Generate cutting planes

- Stronger branching decisions
  - Impose global decisions first (e.g., on the number of vehicles used)
  - Impose very local decisions last (e.g., on the flow on an arc)

Large number of BB nodes

- Use the strong branching strategy
  - Evaluate (maybe heuristically) the lower bounds yielded by several candidate branching variables
  - Choose the variable that led to the best increase (e.g., max min) in the lower bounds

Column generation heuristics: Motivations

- Very large-scale problems
- Find good integer solutions rapidly
- Assess quality of heuristic solutions produced by a very fast heuristic
- Show that an exact method can be adapted for solving large problems
- Local optimization in metaheuristics
  - Matheuristics

Column generation heuristics – basic ingredients

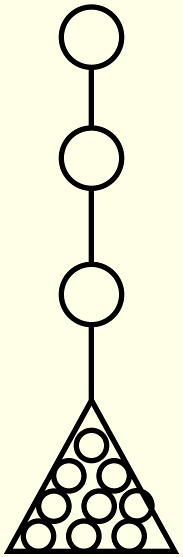
- Linear relaxations
  - Early termination
    - Stop when the progress in the objective function value seems to be stalling
  - Approximate pricing problems
    - Reduced number of variables (arcs)
  - Heuristics for solving the pricing problems
    - Heuristic dominance rule
    - Local search or metaheuristic

Column generation heuristics – basic ingredients

- Branch-and-bound procedure
  - Restricted master problem heuristic (if integrality on the MP variables)
    - Solve the first linear relaxation
    - Solve a MIP on the generated set of columns (no additional columns generated)
  - Diving heuristic
    - Generate columns after each decision
    - No backtracking when feasibility is not hard to reach
    - Multiple decisions at each branching node (as long as they all seem good)

Column generation heuristics – basic ingredients

- To get better quality solutions or to reach feasibility, complete subtree near the bottom



Typical computational results

- Spliet, R. and Desaulniers, G. (2013). The discrete time window assignment vehicle routing problem, submitted to European Journal of Operational Research.

- DTWAVRP definition
  - Goal: Assign a time window to each customer from a subset of potential time windows
  - Consider a set of demand scenarios
  - For each scenario, compute a set of vehicle routes that respect the chosen time windows
  - Objective: Minimize the expected routing costs

DTWAVRP MP model

(2): Assign a time window *w* to each customer *v*
(3): For each scenario ω and each customer *v,* find a route *r* that visits this customer in its selected time window

$$\min \sum_{\omega \in \Omega} p^{\omega} \sum_{r \in R(\omega)} c_r x_r^{\omega} \tag{1}$$

$$\text{s.t.} \quad \sum_{w \in W_v} y_{vw} = 1 \qquad \forall v \in V' \tag{2}$$

$$\sum_{r \in R(\omega)} e_{vwr} x_r^{\omega} = y_{vw} \qquad \forall v \in V', \forall w \in W_v, \forall \omega \in \Omega \tag{3}$$

$$x_r^{\omega} \in \{0,1\} \qquad \forall \omega \in \Omega, \forall r \in R(\omega) \tag{4}$$

$$y_{vw} \in \{0,1\} \qquad \forall v \in V', \forall w \in W_v \tag{5}$$

6 solution methods

- Exact branch-price-and-cut
- Restricted master problem heuristic
  - Integrality only on the TW variables
- Two diving heuristics: 1) With heuristic and exact pricing; 2) Only tabu search pricing
  - Integrality only on the TW variables
- Two rounding heuristics: 1) With heuristic and exact pricing; 2) Only tabu search pricing
  - Solve the linear relaxation and, for each customer, pick the TW with the largest value

Results for small instances

		Exact		Restricted master		TWDiving		TWDiving-Tabu		TWRounding		TWRounding-Tabu	
Inst.	$\|V'\|$	Opt.	Time	Value	Time	Value	Time	Value	Time	Value	Time	Value	Time
11	15	23.04	507.00	23.06	20.42	23.21	99.56	23.04	13.30	25.28	26.49	24.09	6.97
12	15	25.27	5.91	25.27	6.45	25.27	71.19	25.27	8.04	27.06	17.60	26.28	5.15
13	15	22.12	15.92	22.23	26.96	22.34	63.36	22.22	11.52	24.70	16.91	22.51	5.54
14	15	18.46	78.51	18.46	17.55	18.46	141.38	18.46	7.69	19.85	36.83	18.51	5.87
15	15	24.87	1364.49	25.24	21.31	25.33	67.31	24.99	13.69	26.16	23.84	25.17	7.30
16	15	19.82	70.60	20.09	46.42	19.82	491.29	19.82	17.62	22.38	209.38	20.16	11.26
17	15	21.96	501.76	22.05	50.82	22.10	206.31	22.10	13.18	22.05	81.67	24.74	6.46
18	15	22.93	66.83	22.93	16.20	23.07	109.91	22.93	8.95	23.16	28.49	23.57	4.37
19	15	23.14	32.93	23.17	15.56	23.26	65.62	23.34	12.09	23.83	19.91	23.34	6.61
20	15	18.84	35.03	18.95	16.09	19.11	130.55	19.11	17.53	19.86	50.59	19.59	8.18
21	20	27.99	32.94	27.99	27.95	28.14	211.34	28.01	22.24	29.84	35.66	29.95	10.73
22	20	25.63	1225.18	25.63	200.91	25.87	1875.65	25.65	26.41	29.20	349.99	26.07	13.45
23	20	26.53	3079.46	26.53	59.89	26.68	1770.08	26.64	29.64	28.62	139.21	28.48	13.99
24	20	32.36	594.39	32.53	936.09	32.60	301.94	32.99	32.26	32.87	41.64	33.87	12.68
25	20	28.84	105.65	28.84	60.65	29.06	216.76	29.08	29.25	33.64	62.15	29.09	9.91
26	20	26.99	138.12	27.08	49.79	27.23	533.77	27.23	29.92	28.08	16.58	27.53	9.63
27	20	-	3600.00	28.02	552.23	27.65	2989.60	27.93	32.66	28.84	496.24	30.06	12.59
28	20	26.53	3284.31	26.65	108.63	26.58	542.25	26.53	27.88	26.83	79.20	29.41	12.67
29	20	29.49	343.38	29.79	315.37	29.85	549.26	29.75	28.72	31.67	100.89	30.81	9.77
30	20	23.55	2425.43	23.98	184.84	23.59	232.30	23.65	39.32	24.61	35.29	27.60	17.19
31	25	35.47	820.01	35.60	849.26	35.47	601.19	35.84	44.99	36.88	93.98	36.76	18.30
32	25	-	3600.00	32.66	362.23	32.80	530.27	33.17	54.78	36.17	133.68	34.75	25.94
33	25	-	3600.00	31.91	3071.41	31.74	3279.53	31.82	65.17	32.86	431.08	32.19	26.35
34	25	-	3600.00	34.20	255.60	34.14	534.75	34.19	45.59	35.02	92.28	34.79	14.21
35	25	-	3600.00	30.29	266.38	30.29	1138.92	30.29	51.93	31.08	221.22	32.12	26.43
36	25	-	3600.00	32.68	2062.42	33.00	1368.55	32.55	36.50	34.36	127.69	34.50	19.11
37	25	-	3600.00	27.49	635.48	27.81	767.11	27.75	68.48	28.89	207.45	27.66	28.60
38	25	34.83	214.90	34.84	210.75	35.39	746.02	34.93	50.67	37.45	99.64	36.85	18.33
39	25	-	3600.00	34.41	1108.48	34.39	807.19	34.67	56.16	35.87	156.53	36.32	23.34
40	25	30.73	489.06	30.76	231.39	31.30	656.45	30.98	69.26	33.19	133.76	31.27	24.32
41	30		3600.00	36.57	3600.00	36.57	641.78	36.54	85.75	37.72	167.14	37.47	20.60

Results for larger instances

		Exact		Restricted master		TWDiving		TWDiving-Tabu		TWRounding		TWRounding-Tabu	
Inst.	$\|V'\|$	Opt.	Time	Value	Time	Value	Time	Value	Time	Value	Time	Value	Time
41	30	-	3600.00	36.57	3600.00	36.57	641.78	36.54	85.75	37.72	167.14	37.47	29.69
42	30	-	3600.00	41.05	3600.00	41.13	869.45	41.11	88.22	43.77	145.94	42.26	33.92
43	30	-	3600.00	37.59	3600.00	37.54	2013.42	37.48	100.49	39.86	176.64	37.82	26.21
44	30	-	3600.00	38.02	572.11	38.09	647.73	38.40	96.47	39.68	101.68	38.33	31.68
45	30	-	3600.00	37.58	3600.00	36.96	1003.62	37.08	96.00	38.69	127.13	38.48	25.37
46	30	-	3600.00	35.25	3600.00	-	3600.00	35.01	105.84	35.82	587.70	35.62	37.46
47	30	-	3600.00	42.49	3600.00	42.68	425.35	42.59	61.90	43.69	92.93	44.54	25.93
48	30	-	3600.00	37.31	3600.00	-	3600.00	37.09	95.03	39.70	484.51	37.65	35.19
49	30	-	3600.00	41.15	3600.00	40.99	1820.52	40.86	75.30	42.48	221.60	43.52	22.12
50	30	-	3600.00	40.32	3600.00	39.90	1285.75	40.05	98.92	42.44	180.35	41.88	28.07
51	40	-	3600.00	41.95	3600.00	41.54	3078.58	41.82	249.52	43.26	592.77	42.93	69.23
52	40	-	3600.00	54.82	3600.00	-	3600.00	47.99	234.57	50.02	943.37	49.67	65.26
53	40	-	3600.00	46.51	3600.00	-	3600.00	41.80	273.46	42.95	422.37	43.61	70.70
54	40	-	3600.00	46.33	3600.00	-	3600.00	45.96	247.84	47.79	752.28	47.18	64.43
55	40	-	3600.00	-	3600.00	-	3600.00	48.39	207.66	49.01	531.84	51.19	57.68
56	40	-	3600.00	-	3600.00	-	3600.00	44.65	295.71	46.12	2706.25	45.55	88.90
57	40	-	3600.00	-	3600.00	-	3600.00	44.28	230.39	45.69	896.51	46.00	57.80
58	40	-	3600.00	43.28	3600.00	43.21	3234.82	43.25	193.02	43.77	584.46	43.64	59.48
59	40	-	3600.00	49.18	3600.00	-	3600.00	48.76	195.76	49.69	634.14	51.77	67.10
60	40	-	3600.00	47.76	3600.00	-	3600.00	47.62	200.10	47.58	395.48	50.34	46.93
61	50	-	3600.00	-	3600.00	-	3600.00	52.18	439.03	53.28	1210.39	53.61	87.25
62	50	-	3600.00	-	3600.00	-	3600.00	55.84	402.90	58.01	1026.38	58.07	96.25
63	50	-	3600.00	-	3600.00	-	3600.00	50.38	517.14	51.79	1065.61	51.50	127.64
64	50	-	3600.00	-	3600.00	-	3600.00	51.25	534.22	55.62	1823.49	53.72	121.79
65	50	-	3600.00	-	3600.00	-	3600.00	54.40	446.72	56.37	1062.77	55.55	108.89
66	50	-	3600.00	-	3600.00	-	3600.00	56.95	514.13	60.03	2003.95	59.05	99.68
67	50	-	3600.00	58.29	3600.00	-	3600.00	57.67	404.41	59.27	679.43	58.55	99.97
68	50	-	3600.00	-	3600.00	-	3600.00	55.79	510.41	58.89	1302.23	56.74	147.42
69	50	-	3600.00	-	3600.00	-	3600.00	53.64	374.07	55.40	1478.90	56.65	104.94
70	50	-	3600.00	-	3600.00	-	3600.00	56.80	432.60	57.71	1110.55	58.04	95.30
71	60	-	3600.00	-	3600.00	-	3600.00	63.85	834.21	-	3600.00	65.93	168.69
72	60	-	3600.00	-	3600.00	-	3600.00	62.18	765.56	-	3600.00	64.48	194.36
73	60	-	3600.00	-	3600.00	-	3600.00	64.79	714.06	-	3600.00	66.12	161.60
74	60	-	3600.00	-	3600.00	-	3600.00	68.84	703.17	70.91	3115.44	70.78	131.63
75	60	-	3600.00	-	3600.00	-	3600.00	63.65	707.01	64.49	2115.73	65.25	138.44
76	60	-	3600.00	-	3600.00	-	3600.00	64.45	854.16	-	3600.00	66.34	256.01
77	60	-	3600.00	-	3600.00	-	3600.00	61.35	814.93	63.91	2765.95	63.37	223.91
78	60	-	3600.00	-	3600.00	-	3600.00	64.02	772.99	66.03	2531.43	66.09	134.10
79	60	-	3600.00	-	3600.00	-	3600.00	65.40	628.39	69.10	2460.19	66.94	164.99
80	60	-	3600.00	-	3600.00	-	3600.00	64.20	700.72	66.23	1827.12	66.14	156.30

Column-generation-based large neighborhood search heuristics

- Large neighborhood search:
  - Compute a feasible solution
  - While max number of iterations not reached do
    - Destroy part of the current solution
    - Fix the rest of the solution
    - Rebuild a complete solution using a heuristic
- Remarks:
  - Accept worst solutions
  - Here, we use a column generation heuristic for reoptimization

Column-generation-based large neighborhood search heuristic

- Example: Multi-depot vehicle scheduling problem (MDVSP)

  - PEPIN, A.-S., DESAULNIERS, G., HERTZ, A., HUISMAN, D. A comparison of five heuristics for the multiple depot vehicle scheduling problem. *Journal of Scheduling* 12(1), 17-30 (2009).

MDVSP

- n tasks, k depots
- Starting time a_i and duration d_i
- Traveling time t_{ij} from i to j
- A path starts and ends at the same depot
- Task j can follow task i if $a_i+d_i+t_{ij} \leq a_j$
- Limited number of vehicles v_k at each depot k
- Traveling cost c_{ij} from i to j
- Find minimum-cost vehicle schedules such that each task is performed once

Column generation

- One pricing problem for each depot
- Shortest path problem solved by dynamic programming
- Restricted master problem solved by primal simplex algorithm (500 tasks) or barrier algorithm (1000-1500 tasks)
- Algorithm stopped at each branch-and-bound node when progress in the objective function value is insufficient

Heuristic branching: Diving

- At each branching node, path flow variables may be fractional
- Column fixing : fix to 1 columns with flow $\geq$ 0.7 (if none, fix the one with the largest value)
- No backtracks

Initial solution

- Merge all k depots into one where the cost from the new depot d to any task i is

$$c_{di} = \min_{k \in K} c_{ki} \quad \text{and} \quad c_{id} = \min_{k \in K} c_{ik}$$

- Solve SDVSP with simplex algorithm
- Greedy reassignment of paths to the original depots using distances from depots to first and last tasks

Large neighborhood search

- Start with an initial feasible solution

- At each iteration, choose V paths from the current solution
- V = 30 (500 and 1000 tasks), V = 40 (1500 tasks)

- Tasks included in those paths define a small MDVSP
- They are reoptimized by column generation

Choice of paths to reoptimize

- First path randomly selected from a non-tabu list

- Other V-1 paths are either
  - Random paths
  - Paths that have tasks close in time and space to those of the previously selected paths
  - Less frequently chosen paths

- Each strategy has a weight (updated when selected) and a probability to be chosen

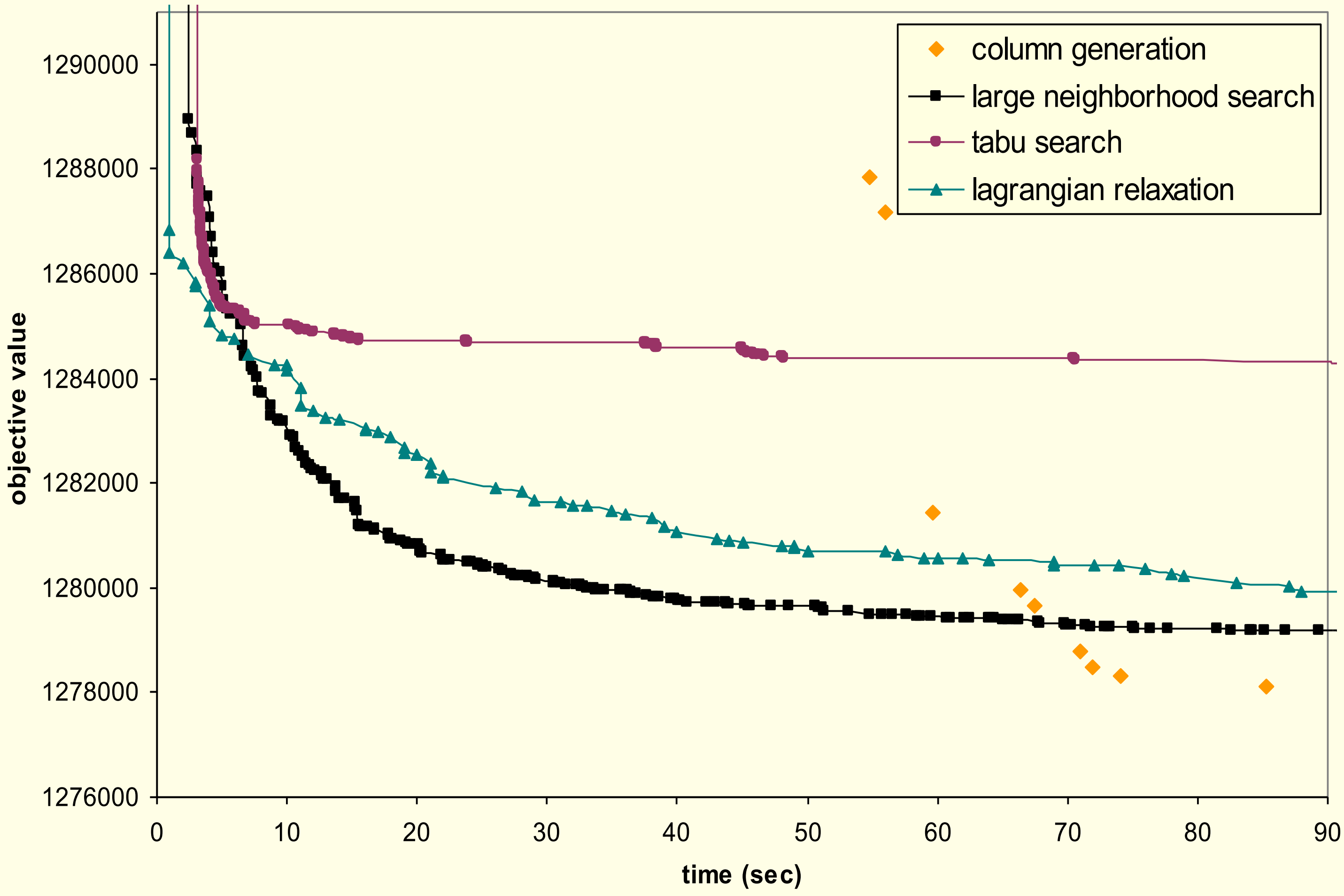
Two other heuristics

- Lagrangean relaxation heuristic
- Tabu search heuristic

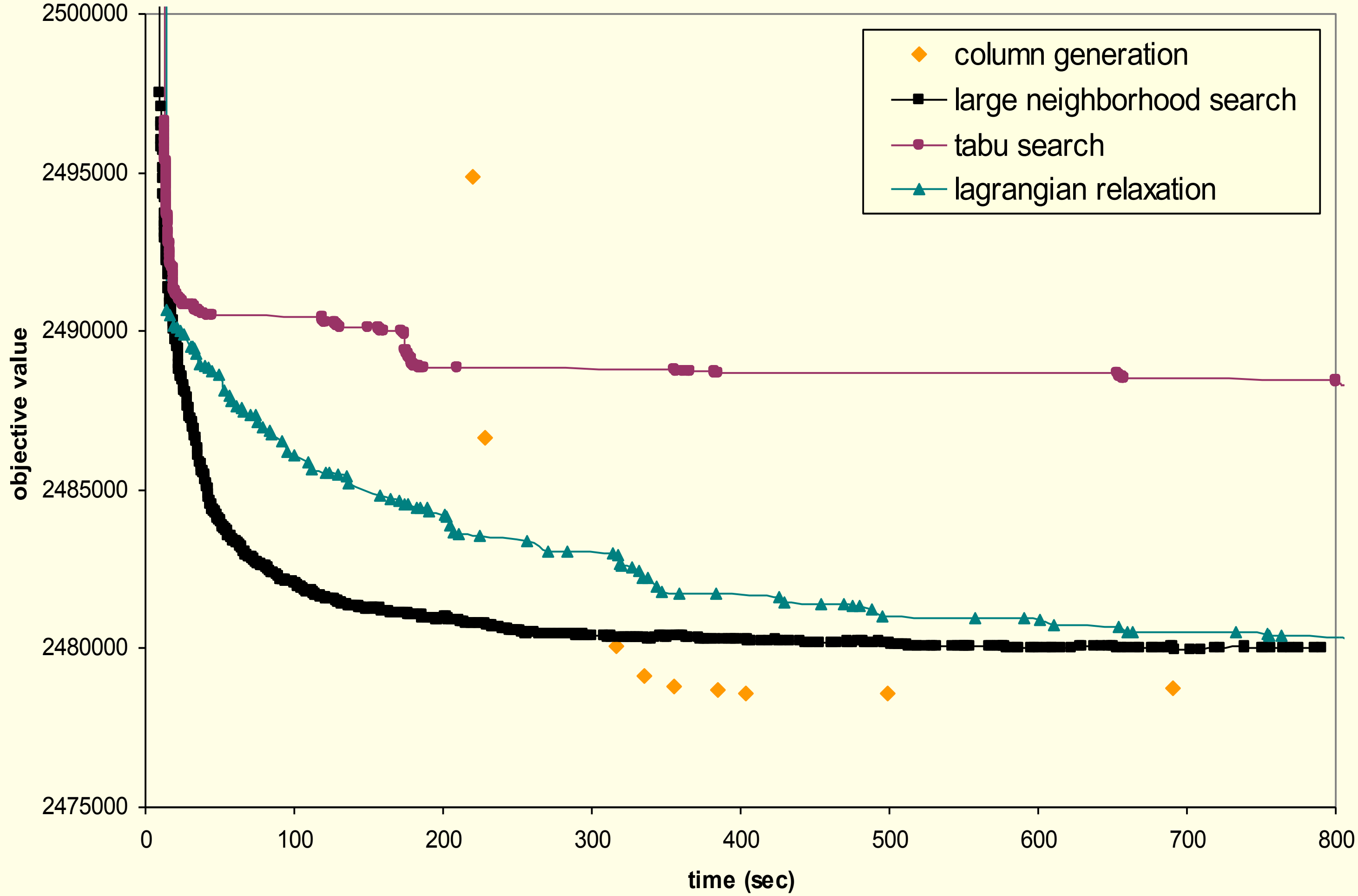
Test features

- 4 depots with 500, 1000, 1500 tasks
- Instances randomly generated as in Carpaneto et al. (1989)
- Averages over 5 instances

Results for 500-task instances



Results for 1000-task instances



Results for 1500-task instances