VRP combined with packing constrints

David Pisinger

VRP with packing constraints

- Demand for integrated VRP and packing
- Two very difficult $\mathcal{NP}$-hard problems
- Only recently studied
- Additional constraints can be imposed

Introduction

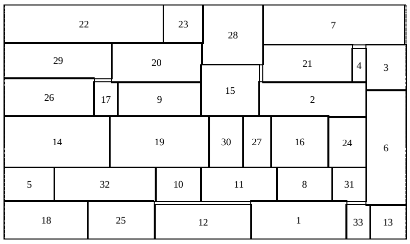
- Capacity constraint $\sum d_i \leq C$ not sufficient
- Complex packing constraints
- Unloading of items should not take too much time

Constraints

- All cargo to one customer located next to each other
- All cargo can be unloaded withouth moving cargo to other customers

Two strategies

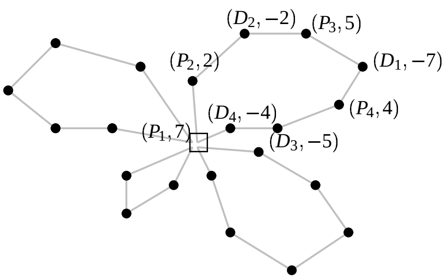
- Forbid unnecessary reorderings
- Punish reorderings



One-, two- and three-dimensional variants

PDP with one-dimensional loading constraint

(Note: CVRP is not interesting with 1D constraint)



- Each vehicle can be considered as a stack
- Pickup: item is added to top of stack
- Delivery: unload sufficiently many items to get item
- Allowed to unload more items
- Unloaded items may be placed in any order to top of stack

Minimize

- Weighted sum of distance and push/pop operations

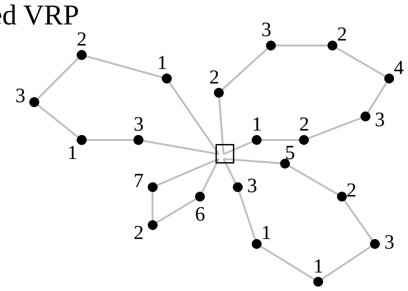
PDP with one-dimensional loading constraint

Very difficult to describe mathematically

- $\mathcal{NP}$-hard when reorderings are punished in objective (reduction from Pickup-Delivery Problem)
- $\mathcal{NP}$-hard when reorderings forbidden (reduction from Bin Packing Problem)

CVRP with two-dimensional loading constraints

Capacitated VRP



$C = 17$

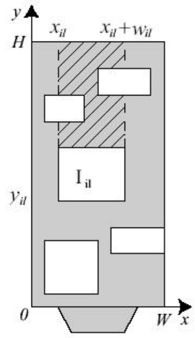
Cargo is packed on *pallets*

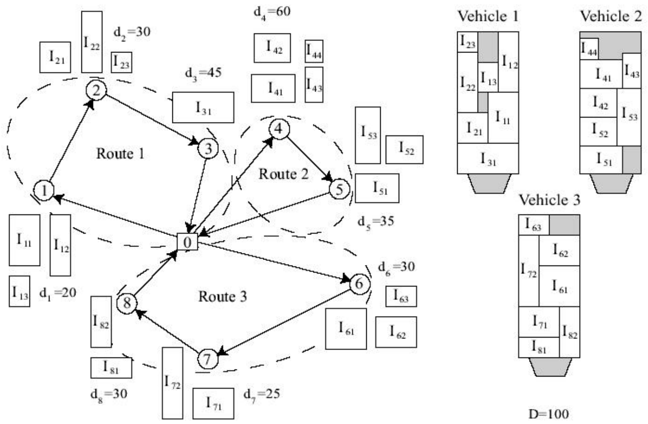
- each customer j has weight demand d_j
- demand takes up rectangles $(w_j^1, h_j^1), \ldots, (w_j^m, h_j^m)$
- rotations of rectangles is *not allowed*
- each vehicle has weight capacity C
- each vehicle has size W, H

Constraints

- The demand of the customers does not exceed C for each vehicle
- All rectangles can be packed
- Cargo for customer j can be unloaded without moving other items

Capacitated VRP



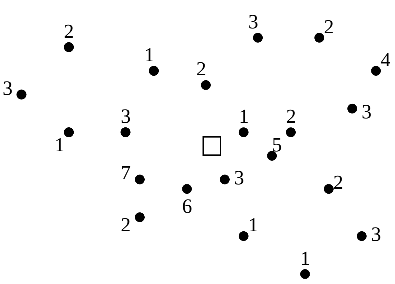


Capacitated VRP

Minimum number of vehicles to service customers S

- Vehicle has capacity C
- Customer i has demand $d_i \geq 0$
- Let $r(S)$ be min number of vehicles needed to service customers in S
- Sometimes we use

$$r(S) = \lceil d(S)/C \rceil$$



$C = 17$

Capacitated VRP

Two-index formulation (asymmetric problem)

- $x_{ij} = 1$ iff edge (i, j) is traversed

Model

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij}$$

Subject to:

$$\sum_{i \in V} x_{ij} = 1 \qquad \forall j \in V \setminus \{0\}$$
$$\sum_{j \in V} x_{ij} = 1 \qquad \forall i \in V \setminus \{0\}$$
$$\sum_{i \in V} x_{i0} = K$$
$$\sum_{j \in V} x_{0j} = K$$
$$\sum_{i \notin S} \sum_{j \in S} x_{ij} \geq r(S) \qquad \forall S \subseteq V \setminus \{0\}, S \neq \emptyset$$
$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \qquad \forall S \subseteq V \setminus \{0\}, S \neq \emptyset$$
$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in V$$

$O(n^2)$ variables, exponential number constraints

- Capacity cut: 2D packing
- Subtour elimination explicitly stated

CVRP with two-dimensional loading constraints

Solved through branch-and-cut [Iori et al., 2005].

- Constraints
$$\sum_{i \notin S} \sum_{j \in S} x_{ij} \geq r(S)$$
are added gradually
- Separation routine is 2D loading problem
- Packing problem is easier than 2D knapsack packing due to restrictions on unloading

Packing

- Contour building
- Start from last customer on route
- Pack one customer at a time
- When new customer is considered, space below is “lost”

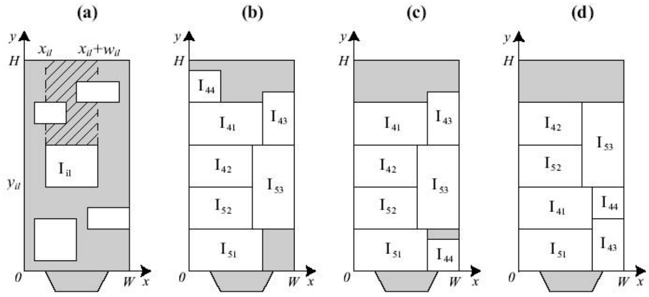
Computational results

- 35 customers and 100 items
- (packing constraint is seldom binding)
- Real-life problems are larger

CVRP with two-dimensional loading constraints

Tabu-search heuristic [Gendreau et al., 2005]

- Unrestricted packing (c,d)
- Sequence packing (a,b)
(each item can be removed through a straight move parallel to the H-edge)



Tabu search sucessfully used for CVRP

- Neighborhood: for each client consider all possible insertions in all routes
- Evaluate cost through GENI procedure
- Penalty if *weight infeasible*
- Penalty if *load infeasible*

Packing constraints

- Weight constraint $\sum_{i \in I} d_i \leq C$ straightforward
- 2D-packing solved heuristically
- If feasible solution found $\rightarrow$ OK
- If no feasible solution $\rightarrow$ may still be possible
- Contour-building approach

CVRP with two-dimensional loading constraints

Tabu search, intensification

- If weight-feasible but load-infeasible, try to pack
- Two-optimization of routes

Computational results

- 255 customers and 255-786 items

Table 2: Details on the instance generation.

Instance	n	*Class 1* M	v	LB	*Class 2* M	v	LB	*Class 3* M	v	LB	*Class 4* M	v	LB	*Class 5* M	v	LB
1) E016-03m	15	15	3	3	24	3	3	31	3	3	37	4	3	45	4	3
2) E016-05m	15	15	5	5	25	5	5	31	5	5	40	5	5	48	5	5
3) E021-04m	20	20	4	4	29	5	4	46	5	4	44	5	4	49	5	4
4) E021-06m	20	20	6	6	32	6	6	43	6	6	50	6	6	62	6	6
5) E022-04g	21	21	4	4	31	4	4	37	4	4	41	4	4	57	5	4
6) E022-06m	21	21	6	6	33	6	6	40	6	6	57	6	6	56	6	6
7) E023-03g	22	22	3	3	32	5	4	41	5	4	51	5	4	55	6	3
8) E023-05s	22	22	5	5	29	5	5	42	5	5	48	5	5	52	6	5
9) E026-08m	25	25	8	8	40	8	8	61	8	8	63	8	8	91	8	8
10) E030-03g	29	29	3	3	43	6	5	49	6	4	72	7	6	86	7	5
11) E030-04s	29	29	4	4	43	6	5	62	7	6	74	7	6	91	7	5
12) E031-09h	30	30	9	9	50	9	9	56	9	9	82	9	9	101	9	9
13) E033-03n	32	32	3	3	44	7	5	56	7	5	78	7	6	102	8	5
14) E033-04g	32	32	4	4	47	7	5	57	7	5	65	7	5	87	8	4
15) E033-05s	32	32	5	5	48	6	5	59	6	6	84	8	7	114	8	6
16) E036-11h	35	35	11	11	56	11	11	74	11	11	93	11	11	114	11	11
17) E041-14h	40	40	14	14	60	14	14	73	14	14	96	14	14	127	14	14
18) E045-04f	44	44	4	4	66	9	7	87	10	8	112	10	8	122	10	6
19) E051-05e	50	50	5	5	82	11	9	103	11	10	134	12	10	157	12	8
20) E072-04f	71	71	4	4	104	14	12	151	15	13	178	16	13	226	16	13
21) E076-07s	75	75	7	7	114	14	12	164	17	14	168	17	14	202	17	14
22) E076-08s	75	75	8	8	112	15	13	154	16	14	198	17	14	236	17	14
23) E076-10e	75	75	10	10	112	14	13	155	16	14	179	16	14	225	16	14
24) E076-14s	75	75	14	14	124	17	14	152	17	14	195	17	14	215	17	14
25) E101-08e	100	100	8	8	157	21	18	212	21	18	254	22	19	311	22	19
26) E101-10c	100	100	10	10	147	19	16	198	20	17	247	20	18	310	20	18
27) E101-14s	100	100	14	14	152	19	17	211	22	19	245	22	19	320	22	19
28) E121-07c	120	120	7	7	183	23	20	242	25	21	299	25	21	384	25	21
29) E135-07f	134	134	7	7	197	24	21	262	26	22	342	28	24	422	28	24
30) E151-12b	150	150	12	12	225	29	25	298	30	27	366	30	27	433	30	27
31) E200-16b	199	199	16	16	307	38	33	402	40	35	513	42	37	602	42	37
32) E200-17b	199	199	17	17	299	38	33	404	39	34	497	39	34	589	39	34
33) E200-17c	199	199	17	17	301	37	32	407	41	35	499	41	36	577	41	36
34) E241-22k	240	240	22	22	370	46	40	490	49	42	604	50	44	720	50	44
35) E253-27k	252	252	27	27	367	45	39	507	50	43	634	50	45	762	50	45
36) E256-14k	255	255	14	14	387	47	41	511	51	44	606	51	44	786	51	44

Table 3: Performance of the Tabu Search heuristic with respect to branch-and-cut. Sequential instances with $n \leq 25$, integer edge costs.

Instance no	*Class*	*Branch-and-cut* z_{bc}		sec_h	sec	*Tabu Search* z_{TS}	sec_h	sec	%gap
1	1	273	*	3.8	3.9	273	0.0	2.4	0.00
	2	285	*	51.4	68.8	285	0.2	4.6	0.00
	3	280	*	17.1	21.4	280	1.3	7.8	0.00
	4	288	*	1.8	2.4	290	0.3	7.5	0.69
	5	279	*	53.1	53.1	279	2.3	15.7	0.00
2	1	329	*	0.2	0.5	329	0.1	1.4	0.00
	2	342	*	11.1	11.9	342	1.1	2.0	0.00
	3	347	*	6.4	8.1	350	0.2	3.2	0.86
	4	336	*	22.4	23.4	336	0.1	6.7	0.00
	5	329	*	29.2	29.4	329	0.3	6.1	0.00
3	1	351	*	14.9	15.6	351	0.2	8.4	0.00
	2	389	*	32.6	65.8	407	8.9	9.8	4.63
	3	387	*	6.0	6.1	387	1.0	20.0	0.00
	4	374	*	39.3	39.4	374	14.3	21.9	0.00
	5	369	*	0.2	0.2	369	1.0	29.9	0.00
4	1	423	*	0.4	0.4	423	0.2	5.7	0.00
	2	434	*	5.3	5.5	434	0.3	7.4	0.00
	3	432	*	5.2	8.2	438	5.0	12.8	1.39
	4	438	*	23.9	26.4	451	0.8	16.9	2.97
	5	423	*	23.0	44.7	423	1.2	27.1	0.00
5	1	367	*	0.1	0.1	367	2.9	12.8	0.00
	2	380	*	8.0	8.4	396	4.0	19.7	4.21
	3	373	*	1.8	1.9	377	0.6	20.5	1.07
	4	377	*	50.2	50.3	406	5.9	33.0	7.69
	5	389	*	2928.2	2928.3	389	5.7	57.8	0.00
6	1	488	*	5.9	10.7	488	0.1	10.3	0.00
	2	491	*	145.8	145.9	498	1.8	14.0	1.43
	3	496	*	135.4	150.4	496	11.2	16.9	0.00
	4	489	*	14.1	16.6	503	0.1	40.2	2.86
	5	488	*	10.8	13.3	488	0.7	34.2	0.00
7	1	558	*	0.0	0.0	558	0.3	22.0	0.00
	2	724	*	32.0	32.5	752	0.7	18.9	3.87
	3	698	*	3.2	4.4	704	19.9	29.8	0.86
	4	714	*	2596.8	2597.3	742	26.5	50.4	3.92
	5	742	*	738.9	747.2	743	16.1	75.1	0.13
8	1	657	*	0.0	0.0	657	7.0	31.3	0.00
	2	720	*	75.9	91.8	720	3.1	20.1	0.00
	3	730	*	70.0	73.0	752	20.5	33.4	3.01
	4	701	*	7.4	14.1	722	11.2	50.0	3.00
	5	721	*	1128.9	1128.9	736	12.0	90.2	2.08
9	1	609	*	6.2	31.91	609	1.9	11.9	0.00
	2	612	*	453.5	460.52	612	2.9	15.4	0.00
	3	615	*	164.8	194.31	626	7.9	38.4	1.79
	4	626	*	852.1	1593.34	627	5.9	43.5	0.16
	5	609	*	47.4	69.17	609	8.5	81.9	0.00

Table 4: Performance of the Tabu Search heuristic with respect to branch-and-cut. Sequential instances with $25 < n \leq 40$, integer edge costs.

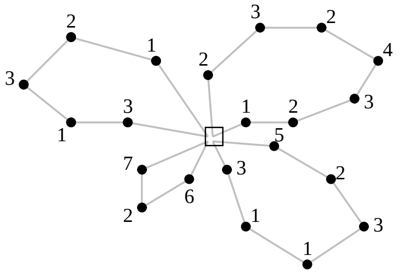
Instance no	*Class*	*Branch-and-cut* z_{bc}		sec_h	sec	*Tabu Search* z_{TS}	sec_h	sec	%gap
10	1	524	*	1226.2	83249.5	544	19.9	97.3	3.82
	2	774		48373.3	86401.1	703	3.8	72.2	−9.17
	3	638	*	2304.5	3150.0	676	47.9	118.7	5.96
	4	738	*	12671.3	12696.3	773	47.3	156.9	4.74
	5	706		48220.2	70308.0	724	84.5	308.9	2.55
11	1	500	*	0.1	0.1	500	0.8	107.8	0.00
	2	789		99.8	86400.5	734	4.7	72.2	−6.97
	3	763		63747.8	86400.3	785	72.2	101.7	2.88
	4	881		85181.5	86400.8	877	196.4	209.5	−0.45
	5	695		64392.9	86400.1	696	279.2	387.0	0.14
12	1	599		50093.5	86400.3	598	19.4	33.8	−0.17
	2	625		75.6	86400.7	628	9.6	42.9	0.48
	3	597		36171.0	86400.6	597	12.9	50.7	0.00
	4	624		86321.6	86400.2	640	96.0	120.6	2.56
	5	602		22352.4	86400.4	597	103.5	188.2	−0.83
13	1	1991	*	13.2	14.5	1991	47.4	218.7	0.00
	2	3523		10784.1	86400.3	2775	43.7	123.1	−21.23
	3	2570	*	3087.9	32124.5	2696	121.5	170.3	4.90
	4	2673		34999.2	86400.4	2743	29.3	277.0	2.62
	5	2807		83628.4	86400.1	2737	237.5	691.9	−2.49
14	1	827		25734.8	86400.4	823	21.8	145.0	−0.48
	2	1459		80521.1	86400.8	1266	52.3	144.4	−13.23
	3	1211		42204.7	86400.1	1204	137.9	207.1	−0.58
	4	1166	*	25391.4	25542.6	1187	108.5	324.9	1.80
	5	1504		80511.1	86400.1	1309	103.2	895.0	−12.97
15	1	907	*	17.8	17.8	907	51.9	196.4	0.00
	2	1203		386.9	86401.4	1135	94.7	133.9	−5.65
	3	1405		15880.7	86401.2	1183	37.1	205.9	−15.80
	4	1358		55614.5	86400.3	1372	268.2	332.2	1.03
	5	1390		59867.8	86400.1	1361	651.4	671.7	−2.09
16	1	682	*	7086.5	7086.7	682	56.1	96.6	0.00
	2	682	*	1767.0	3374.8	682	20.5	91.7	0.00
	3	682	*	345.1	3853.3	682	15.3	138.1	0.00
	4	691	*	33375.9	33391.8	704	21.7	206.1	1.88
	5	682	*	339.0	2784.0	682	62.8	276.6	0.00
17	1	859		493.1	86400.3	842	84.4	158.6	−1.98
	2	866		283.8	86400.4	851	56.6	132.5	−1.73
	3	850		656.0	86400.0	842	38.8	200.6	−0.94
	4	853		102.9	86400.4	845	7.4	279.7	−0.94
	5	845		263.6	86400.5	842	44.7	402.2	−0.36

Table 6: Performance of the Tabu Search heuristic. Real edge costs.

Inst.	*CVRP* z_1	sec_h	sec	*Unrestricted* $\overline{z}_{2-5}$	sec_h	sec	*Sequential* $\overline{z}_{2-5}$	sec_h	sec
1	278.73	2.0	2.2	291.60	4.2	9.7	299.09	2.6	9.2
2	334.96	0.0	1.4	341.02	0.1	3.7	345.23	0.4	3.5
3	359.77	3.5	8.4	377.35	1.6	19.6	385.30	3.8	18.9
4	430.88	0.1	5.7	437.45	0.5	14.7	443.42	1.4	17.0
5	375.28	1.4	12.8	380.20	5.0	25.2	384.06	4.1	27.6
6	495.85	0.3	8.7	501.02	7.2	18.8	502.78	5.1	19.5
7	568.56	0.5	22.6	700.34	6.3	48.4	721.90	15.5	53.0
8	568.56	0.5	36.2	694.99	11.2	61.3	722.73	32.8	83.7
9	607.65	0.4	13.5	619.69	3.6	41.2	624.06	10.6	40.0
10	538.79	6.1	81.7	700.39	36.0	192.0	714.90	43.5	179.6
11	505.01	2.5	98.9	739.04	55.7	207.6	773.45	99.0	199.4
12	610.57	28.5	32.5	620.62	49.0	82.7	631.85	58.8	99.5
13	2006.34	29.9	161.6	2598.20	57.5	332.0	2687.03	49.0	312.8
14	837.67	22.2	152.1	1047.72	375.8	565.6	1101.49	146.0	439.5
15	837.67	1.7	182.5	1201.38	156.7	375.9	1240.89	165.4	313.4
16	698.61	2.7	99.0	702.03	20.5	160.1	704.85	28.0	157.2
17	862.62	59.0	162.8	866.37	64.9	218.2	866.50	88.9	226.2
18	723.54	81.9	587.3	1085.84	589.3	1318.0	1116.17	566.5	1167.7
19	524.61	128.8	641.9	772.25	633.7	1524.5	802.48	365.2	1521.5
20	241.97	253.6	1005.2	564.67	954.5	3237.3	581.81	808.9	3370.3
21	688.18	325.0	2978.7	1066.21	460.1	3600.0	1110.19	1702.2	3561.2
22	740.66	2070.7	3600.1	1087.46	1191.2	3544.5	1130.33	1573.8	3461.8
23	860.47	2210.1	3600.0	1104.72	2032.4	3538.6	1186.36	675.8	3600.0
24	1048.91	866.9	3600.0	1187.62	1454.1	3256.1	1248.43	2642.5	3324.6
25	830.26	2371.0	3598.8	1436.09	1205.8	3600.0	1480.63	2336.5	3600.1
26	819.56	3597.6	3600.3	1404.49	1173.9	3600.0	1471.74	1554.6	3600.3
27	1099.95	355.9	3600.1	1450.18	521.3	3600.0	1524.22	1308.2	3600.0
28	1078.27	985.2	3600.0	2738.31	2051.2	3600.1	2858.53	2576.9	3600.1
29	1179.01	3080.0	3600.0	2474.33	1406.5	3600.0	2575.28	1162.5	3600.2
30	1061.55	1834.4	3600.8	1948.72	1185.4	3600.3	2076.20	2021.4	3600.2
31	1464.04	288.8	3600.9	2506.99	2375.8	3600.2	2592.17	2102.2	3600.5
32	1352.61	1780.8	3600.1	2486.43	1664.8	3600.4	2605.10	2305.2	3600.5
33	1361.51	2531.7	3601.1	2504.00	1843.2	3600.2	2610.55	2221.2	3600.6
34	858.94	1941.9	3604.4	1466.06	1359.1	3601.3	1546.06	2184.4	3601.0
35	992.86	766.7	3600.3	1765.30	2061.7	3600.4	1985.44	2223.1	3600.2
36	678.87	1530.9	3603.9	1909.88	2265.8	3600.1	1946.66	2626.3	3600.9
Aver.	792.31	772.9	1757.4	1216.08	750.7	1837.4	1266.61	977.3	1831.9

CVRP with three-dimensional loading constraints

Sofa-packing



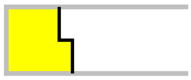
- Capacitated VRP
- Three-dimensional packing problem
- Extremely difficult to solve

Sofa-packing

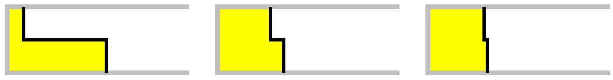
Due to nature of cargo

- All cargo next to each other
- L-shaped packings

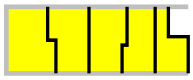
Packing chart of a single customer



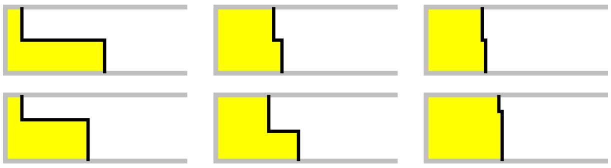
For each customer find set S of undominated solutions



Loading of a whole container



Sofa-packing



- w_{ij} smallest width packing i and j together
- w_{0j} smallest widht packing j alone

We may calculate all w_{ij} in time $O(n^2 S^2)$

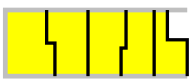
Length of packing with customers $I = \{1, 2, \ldots, m\}$

$$w_{12} + w_{34} + \ldots + w_{m-1,m}$$

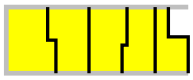
If odd number of customers I

$$\min \begin{cases} w_{12} + w_{34} + \ldots + w_{m-2,m-1} + w_{0m} \\ w_{01} + w_{12} + \ldots + w_{m-1,m} \end{cases}$$

Can be calculated in $O(m)$ time



Sofa-packing



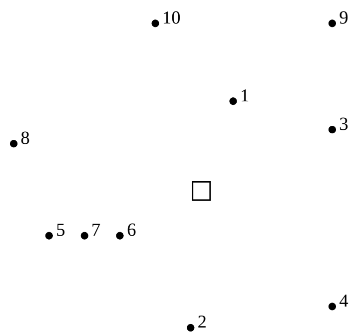
$$H = 100, L = 150$$

- Simple heuristic for VRP problem
- Simulated annealing
- Initially start with 10 routes

Sofa-packing

```
route  1 cost   0: L1   0 L2   0   0 0
route  2 cost  69: L1 101 L2 108   0 1 9 3 0
route  3 cost 133: L1 108 L2 109   0 4 10 2 0
route  4 cost   0: L1   0 L2   0   0 0
route  5 cost   0: L1   0 L2   0   0 0
route  6 cost   0: L1   0 L2   0   0 0
route  7 cost   0: L1   0 L2   0   0 0
route  8 cost  67: L1 109 L2 109   0 7 8 5 6 0
route  9 cost   0: L1   0 L2   0   0 0
route 10 cost   0: L1   0 L2   0   0 0
```

Initial route length 413 final route length 269



Conclusion

- It *is* possible to handle VRP and packing
- Simplify problem from 3D to 2D to 1D
- Packing constraint is seldom binding
- Growing area of research

References

M. Gendreau, M. Iori, G. Laporte, and S. Martello. A tabu search heuristic for the vehicle routing problem with two-dimensional loading constraints. 2005. in preparation.

M. Iori, J.J. Salazar Gonzales, and D. Vigo. An exact approach for the vehicle routing problem with two-dimensional loading constraints. 2005.