Some POPMUSIC Applications in Logistics

Comwell Borupgaard, June 03, 2014

Stefan Voß (stefan.voss@hamburg.de)

Some POPMUSIC Applications in Logistics

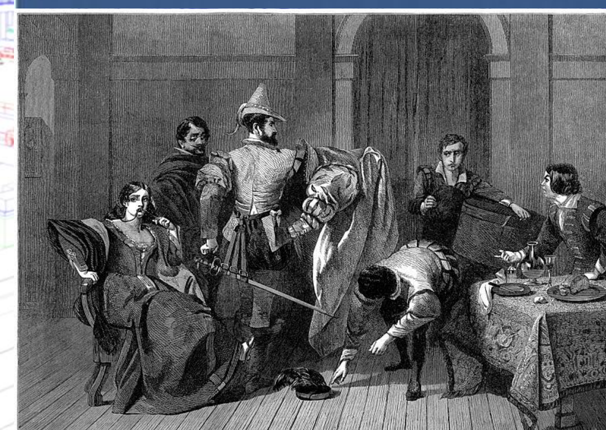
Comwell Borupgaard, June 03, 2014

- The Dynamic Berth Allocation Problem (DBAP)
- A POPMUSIC Approach for the DBAP
- Computational Results
- Conclusions, Future Work, Extensions, New Talks

Stefan Voß (stefan.voss@hamburg.de)

Berth Allocation

- Container terminals are open systems of material flow.
- Main objective: “Minimize the time of the vessel in the port.” (The turnaround time of the container vessels is a main KPI – key performance indicator regarding competitiveness of a maritime container terminal.
- The increasing volume of containers and fierce competition among terminals require an efficient use of the scarce resources in the port.
- The Berth Allocation Problem (BAP) consists of determining a berthing time and a berthing position at the quay for each vessel to be served within a given planning horizon.

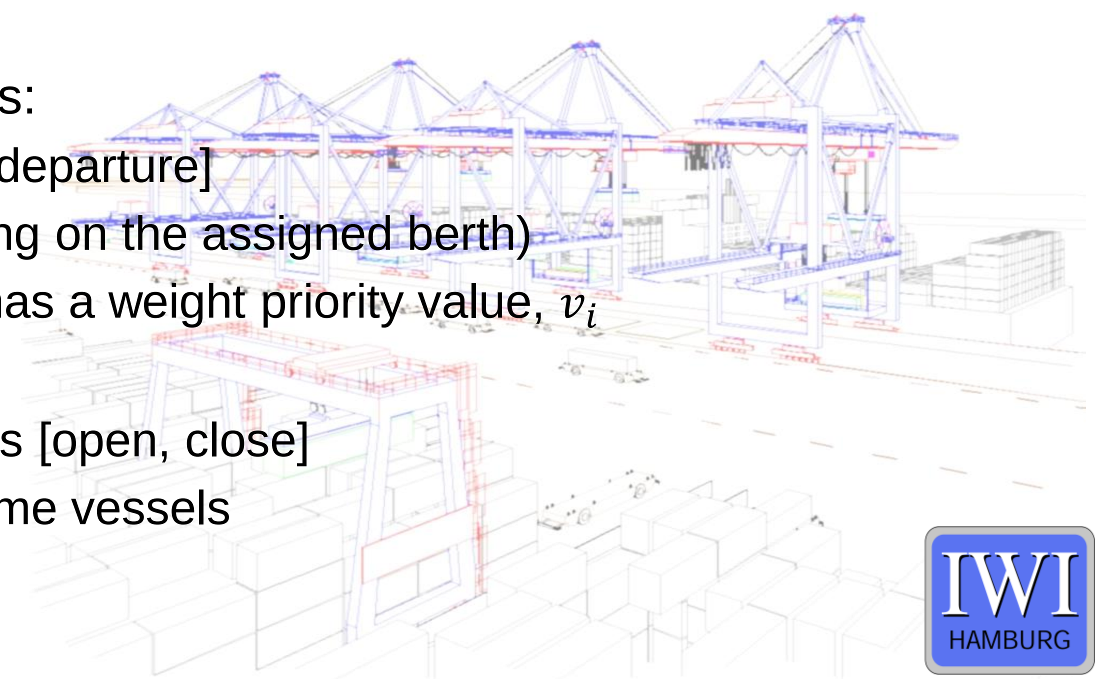


Dynamic Berth Allocation Problem (DBAP)

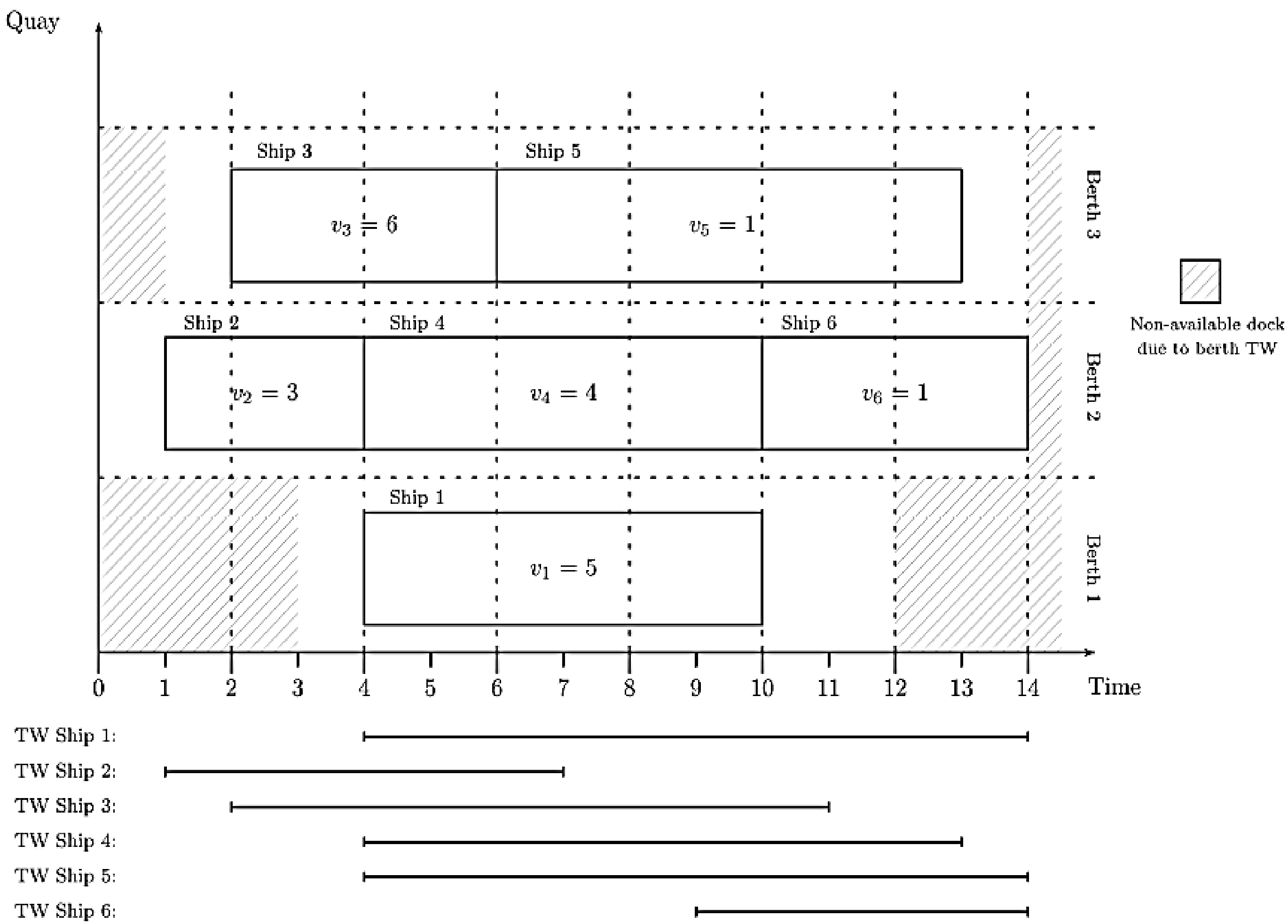
The DBAP aims at determining the berthing position and berthing time for the container vessels arriving at a port, such that the total (weighted) service time for all vessels is minimized.

$$minimize \sum_{i \in N} \sum_{k \in M} v_i \left[T_i^k - a_i + t_i^k \sum_{j \in N \cup d(k)} x_{ij}^k \right]$$

- *NP*-hard (Cordeau et al.)
- *N* arriving container vessels:
  - Time-Windows [arrival, departure]
  - Handling time (depending on the assigned berth)
  - Each container vessel has a weight priority value, v_i
- *M* berths:
  - Availability time-windows [open, close]
  - Forbidden berths for some vessels



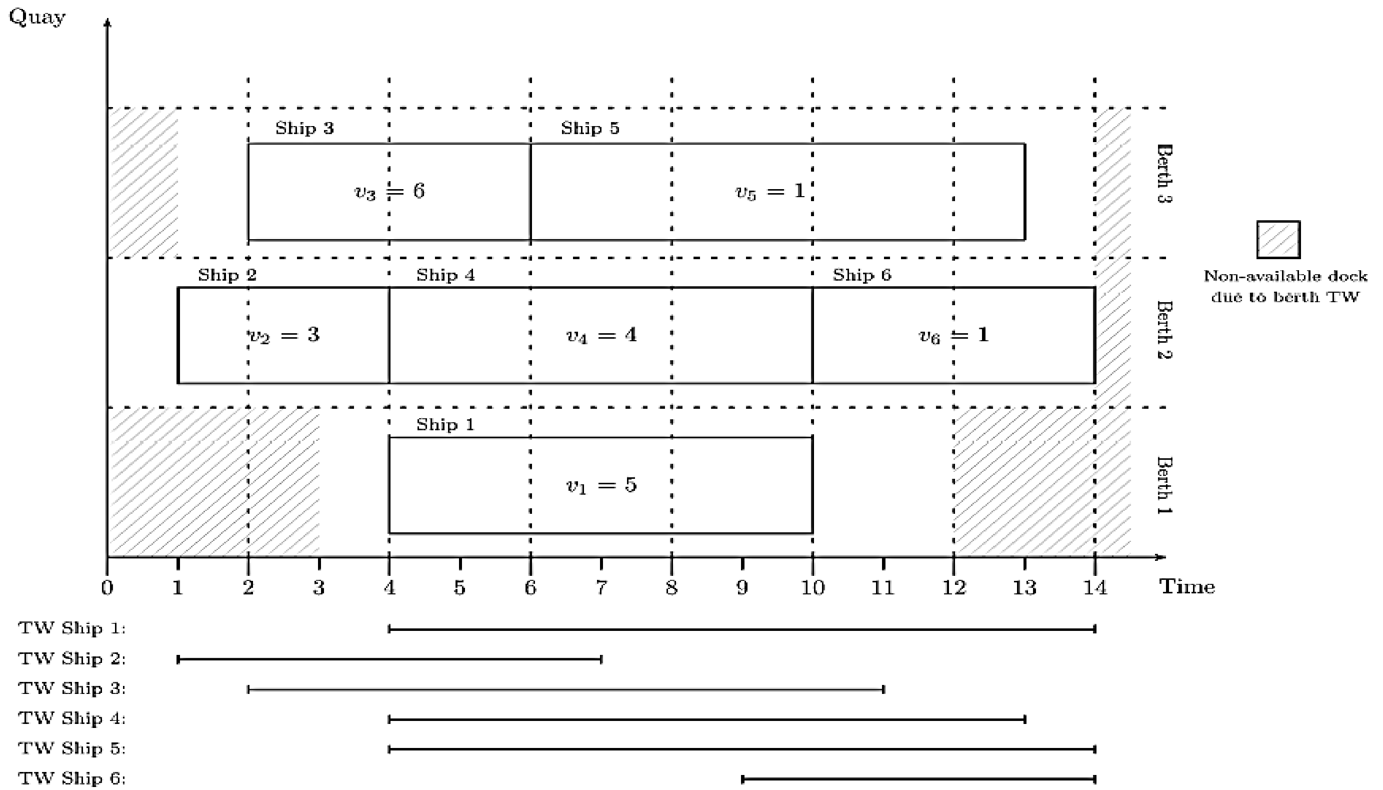
Berth Allocation Problem



Ships handling times depending on the berth

	Berth 1	Berth 2	Berth 3
Ship 1	6	8	5
Ship 2	2	3	4
Ship 3	5	5	4
Ship 4	4	6	5
Ship 5	5	8	7
Ship 6	4	4	5

Berth Allocation Problem



Solution structure for the DBAP

5	0	3	4	1	0	6	1

Move-based vs. Method-based Neighborhoods (Paradigms?)

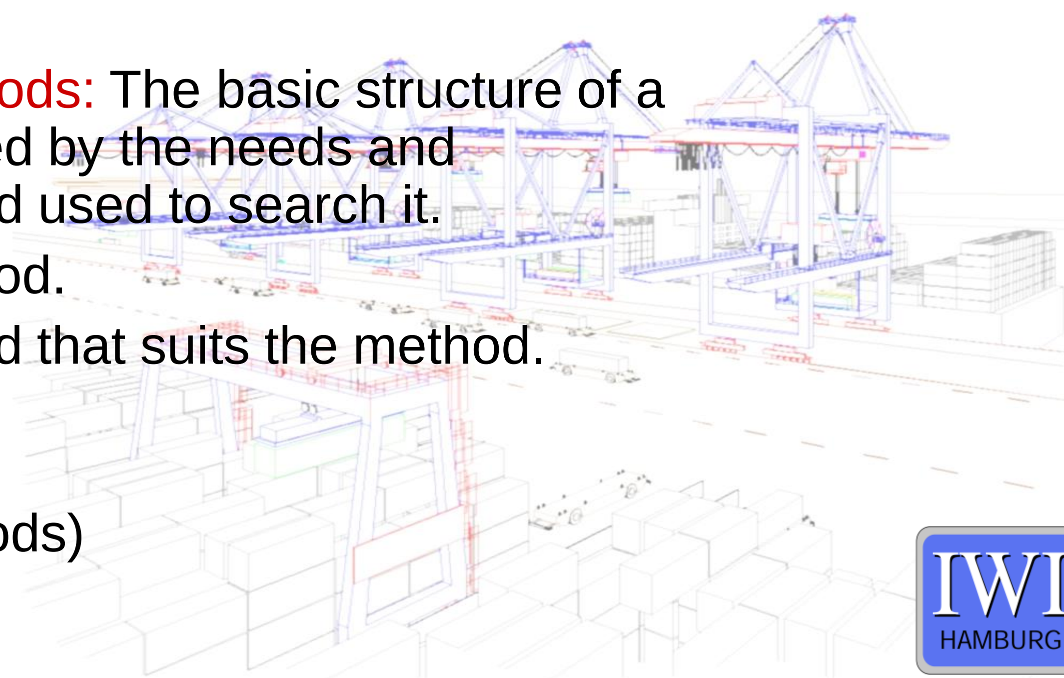
Move-based neighborhoods: Neighborhood search based on topological concepts relating a given solution to similar solutions via local changes or moves. (Example: small homogeneous neighborhoods, very large scale neighborhoods)

1. Define a neighborhood
2. Find efficient methods for its exploration

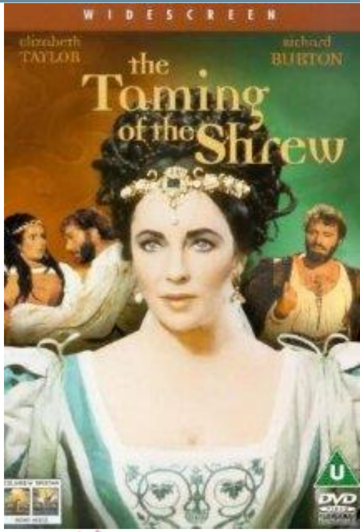
Method-based neighborhoods: The basic structure of a neighborhood is determined by the needs and requirements of the method used to search it.

1. Assume a given method.
2. Define a neighborhood that suits the method.

(Model-based neighborhoods)



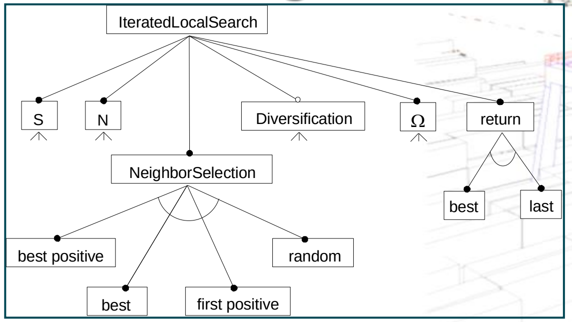
Domain Analysis: Metaheuristics



POPMUSIC

- Taillard/Voß (2002). In: C.C. Ribeiro and P. Hansen (Eds.), *Essays and Surveys in Metaheuristics*, Kluwer, Boston (2002).

Feature Diagrams



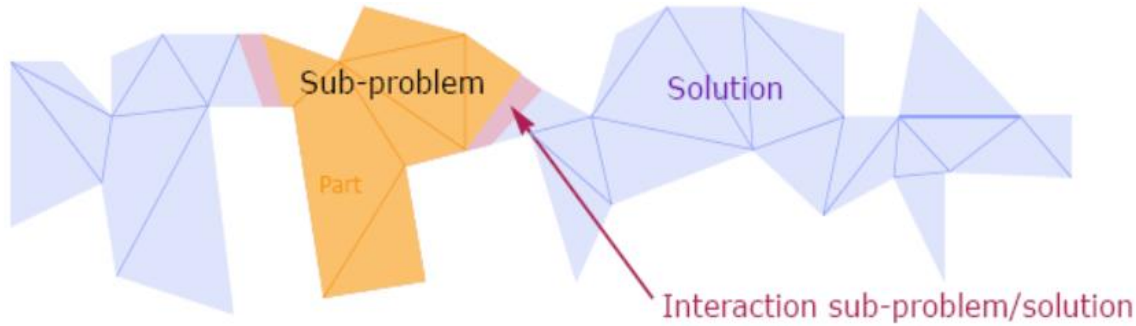
Pseudo Code

IteratedLocalSearch
$< S, N, \mathit{NeighborSelection}, \mathit{Diversification} >$
$(s, T_{\max} = \infty, I_{\max} = \infty, R_{\max} = 1, \omega = \text{false}, \mathit{returnBest} = \text{true})$:

$\Omega : (t \geq T_{\max})$ **or** (ω)

$s_{\text{best}} = s;$
for $r = 1$ **to** $R_{\max}$
  if $r > 1$
    $\mathit{Diversification}(s);$
  $i = 0;$
  do
    $i = i + 1;$
    $s' = \mathit{NeighborSelection} < S, N > (s);$
    if s' is valid
      $s = s';$
      if $f(s) < f(s_{\text{best}})$
        $s_{\text{best}} = s;$
  while $(s'$ is valid$)$ **and** $(i < I_{\max});$
if $\mathit{returnBest}$
  $s = s_{\text{best}};$

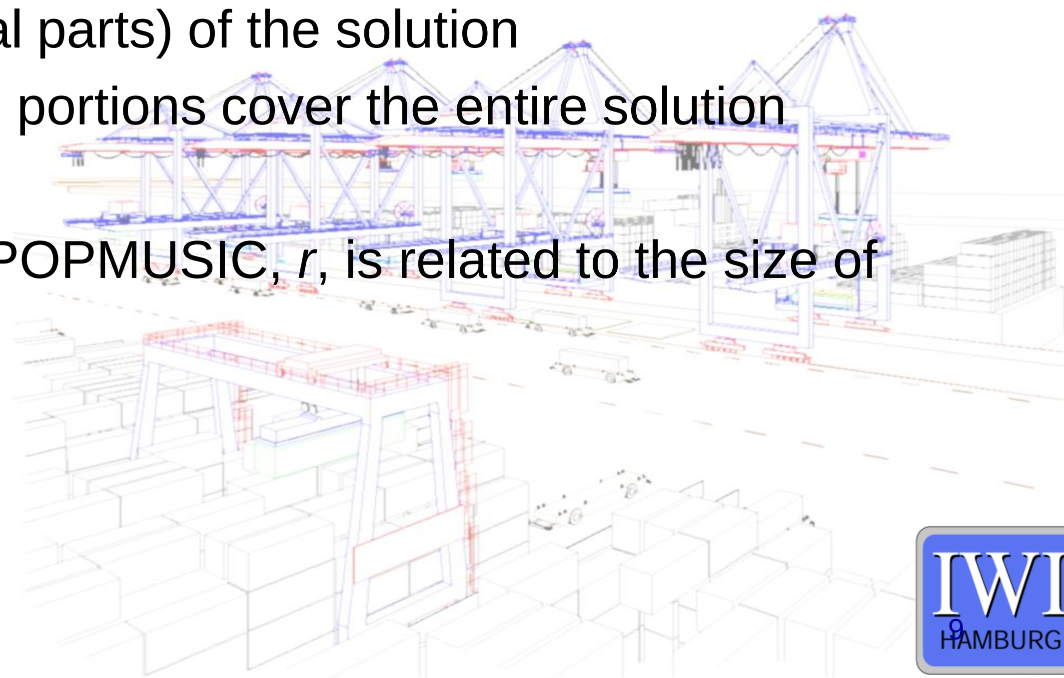
POPMUSIC



General idea:

- Start from an initial solution
- Decompose solution into parts
- Optimize a portion (several parts) of the solution
- Repeat until the optimized portions cover the entire solution

- The unique parameter of POPMUSIC, *r*, is related to the size of the subproblems.



POPMUSIC

Initial Solution:

Generate an initial solution, S, using a Random Greedy method (R-G).

- Berthing order sequence at random
- Assign vessels to berths subject to minimum completion time

Algorithm 2: POPMUSIC approach for the DBAP

Generate an initial solution S at random using R-G
Decompose S in M parts according to the number of berths, $H = \{part_1, ..., part_M\}$
Set $O = \emptyset$
while $O \neq \{part_1, ..., part_M\}$ **do**
  Select a seed part $s_{seed} \in H$ at random
  Build a sub-problem R composed of the r parts of S
  which are the closest to s_{seed}
  Optimize R through solving its GSPP mathematical formulation using a general-purpose solver
  if *R has been improved* **then**
    Update solution S
    $O \leftarrow \emptyset$
  else
    Include s_{seed} in O
return the improved solution S

HAMBURG

POPMUSIC

Decompose S into parts:

5	0	3	4	1	0	6	1

$\boldsymbol{part_1}$ $\boldsymbol{part_2}$ $\boldsymbol{part_3}$

- M = 3 berths
- Make set of optimized parts O empty

Algorithm 2: POPMUSIC approach for the DBAP

```
Generate an initial solution S at random using R-G
Decompose S in M parts according to the number of
  berths, H = {part_1, ..., part_M}
Set O = ∅
while O ≠ {part_1, ..., part_M} do
    Select a seed part s_seed ∈ H at random
    Build a sub-problem R composed of the r parts of
    S
    which are the closest to s_seed
    Optimize R through solving its GSPP
    mathematical formulation using a
    general-purpose solver
    if R has been improved then
        Update solution S
        O ← ∅
    else
        Include s_seed in O
return the improved solution S
```

IWI HAMBURG

POPMUSIC

Sub-problem creation:

- The seed part is selected at random from the *M* parts
- Distance function among the part identifier and its adjacent ones.

r=1

5	0	3	4	1	0	6	1

<u>Sub-problem R</u>

Number of vessels, $N = 4$
Number of berths, $M = 2$
Vessels in the subproblem = {1, 3, 4, 5}
Berths in the subproblem = {1, 2}

Algorithm 2: POPMUSIC approach for the DBAP

```
Generate an initial solution S at random using R-G
Decompose S in M parts according to the number of
berths, H = {part_1, ..., part_M}
Set O = ∅
while O ≠ {part_1, ..., part_M} do
    Select a seed part s_seed ∈ H at random
    Build a sub-problem R composed of the r parts of
    S
    which are the closest to s_seed
    Optimize R through solving its GSPP
    mathematical formulation using a
    general-purpose solver
    if R has been improved then
        Update solution S
        O ← ∅
    else
        Include s_seed in O
return the improved solution S
```

IWI HAMBURG

POPMUSIC

Optimize subproblem R:

- Mathematical formulation for the DBAP proposed by Christensen and Holst
- In case R is improved, then empty the set of optimized parts *O*. If not, include the seed part (POPMUSIC) or all the parts composing R (POPMUSIC-G) into *O*.
- Once no parts can be improved, then the Optimized set is full and POPMUSIC stops.

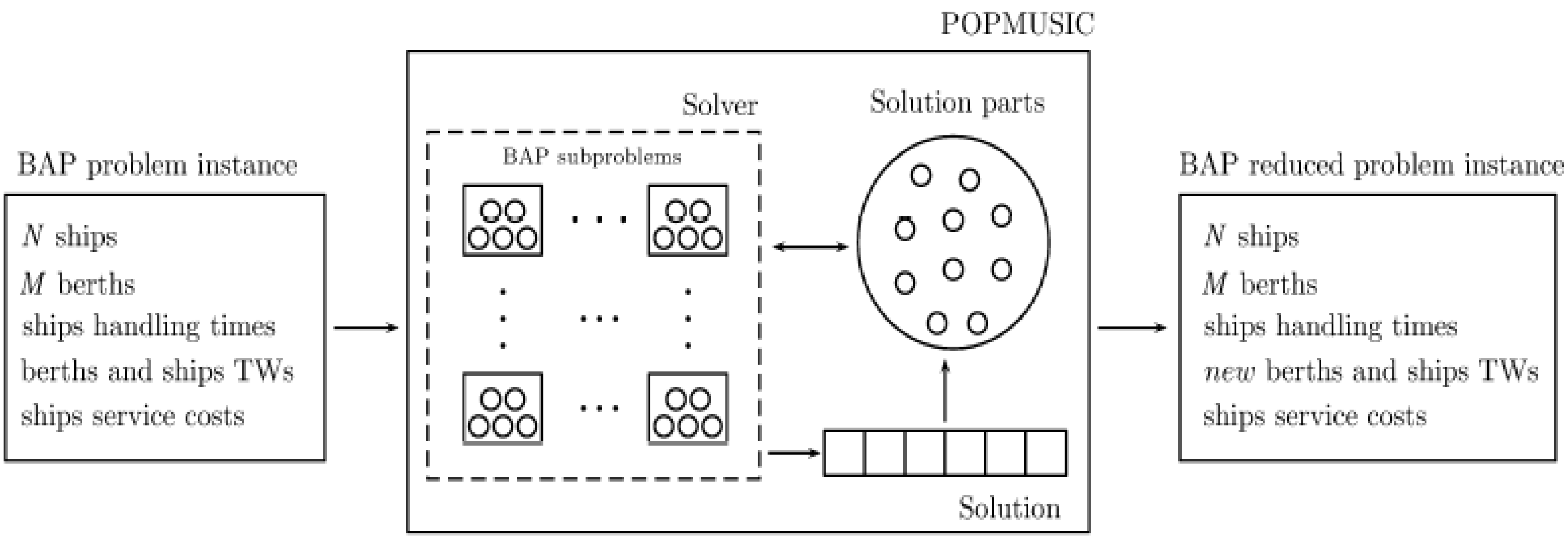
Algorithm 2: POPMUSIC approach for the DBAP

Generate an initial solution S at random using R-G
Decompose S in M parts according to the number of berths, $H = \{part_1, ..., part_M\}$
Set $O = \emptyset$
while $O \neq \{part_1, ..., part_M\}$ **do**
    Select a seed part $s_{seed} \in H$ at random
    Build a sub-problem R composed of the r parts of S
    which are the closest to s_{seed}
    Optimize R through solving its GSPP mathematical formulation using a general-purpose solver
    if *R has been improved* **then**
        Update solution S
        $O \leftarrow \emptyset$
    else
        Include s_{seed} in O
return the improved solution S

IWI HAMBURG

POPMUSIC

As indicated by Lalla et al., the GSPP mathematical formulation runs out of memory for some instances for the DBAP. Those instances have not yet been solved to proven optimality.



Computational Results

joint work with Eduardo Lalla-Ruiz

Instances provided by Cordeau et al. (2005) and Lalla et al. (2012).

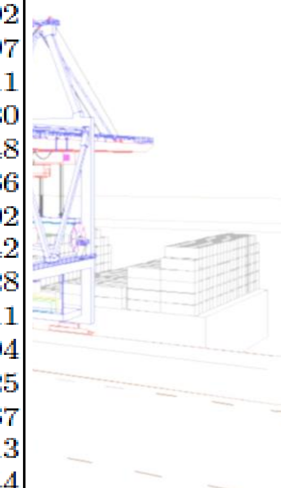
Comparison among:

- GSPP mathematical formulation using CPLEX 12.2 (Xu et al.): PC with Intel Core 3.16 GHz with 4 GB of RAM
- Clustering Search with Simulated Annealing (CS-SA): PC with AMD Athlon 64 with 2.2GHz and 1 GB of RAM
- Particle Swarm Optimization (PSA): PC with Intel Core 2 Duo E8400 3.00 GHz with 2 GB of RAM
- Tabu Search with Path-Relinking (T2S+PR): PC with Intel 3.16 GHz with 4 GB of RAM
- POPMUSIC/POPMUSIC-G using CPLEX 12.2: PC with Intel Core 3.16 GHz with 4 GB of RAM

Computational Results

Instances from Cordeau et al. (2005); joint work with Eduardo Lalla-Ruiz

	GSPP		POPMUSIC		POPMUSIC-G		CS-SA			PSO		
	r=1,2,3,4		r=1,2,3,4		r=1,2,3,4							
	opt.	t(s.)	obj. val	t(s.)	obj. val	t(s.)	obj. val	gap (%)	t(s.)	obj. val	gap (%)	t(s.)
i01	1409	33.20	1409	34.20	1409	11.47	1409	0.00	12.47	1409	0.00	11.11
i02	1261	29.18	1261	54.21	1261	12.61	1261	0.00	12.59	1261	0.00	7.89
i03	1129	28.17	1129	33.79	1129	13.89	1129	0.00	12.64	1129	0.00	7.48
i04	1302	29.20	1302	35.68	1302	13.65	1302	0.00	12.59	1302	0.00	6.03
i05	1207	27.93	1207	28.14	1207	11.57	1207	0.00	12.68	1207	0.00	5.84
i06	1261	29.75	1261	36.60	1261	15.15	1261	0.00	12.56	1261	0.00	7.67
i07	1279	32.89	1279	26.73	1279	12.20	1279	0.00	12.63	1279	0.00	7.50
i08	1299	30.19	1299	57.12	1299	15.56	1299	0.00	12.57	1299	0.00	9.94
i09	1444	30.89	1444	54.20	1444	13.59	1444	0.00	12.58	1444	0.00	4.25
i10	1213	29.14	1213	26.57	1213	12.29	1213	0.00	12.61	1213	0.00	5.20
i11	1368	30.62	1368	22.32	1368	14.84	1368	0.00	12.58	1368	0.00	10.52
i12	1325	28.93	1325	42.51	1325	13.79	1325	0.00	12.61	1325	0.00	12.92
i13	1360	30.14	1360	35.68	1360	13.81	1360	0.00	12.58	1360	0.00	11.97
i14	1233	26.17	1233	41.09	1233	12.92	1233	0.00	12.56	1233	0.00	7.11
i15	1295	26.59	1295	35.87	1295	12.89	1295	0.00	12.61	1295	0.00	8.30
i16	1364	24.65	1364	40.18	1364	13.29	1364	0.00	12.67	1364	0.00	8.48
i17	1283	24.89	1283	27.89	1283	11.84	1283	0.00	13.80	1283	0.00	5.66
i18	1345	23.17	1345	35.88	1345	13.39	1345	0.00	14.46	1345	0.00	8.02
i19	1367	24.66	1367	27.83	1367	13.69	1367	0.00	13.73	1367	0.00	11.42
i20	1328	30.77	1328	23.89	1328	12.94	1328	0.00	12.82	1328	0.00	12.28
i21	1341	24.59	1341	19.21	1341	12.48	1341	0.00	12.68	1341	0.00	7.11
i22	1326	23.76	1326	59.37	1326	13.37	1326	0.00	12.62	1326	0.00	7.94
i23	1266	22.72	1266	38.33	1266	14.36	1266	0.00	12.62	1266	0.00	7.25
i24	1260	21.06	1260	35.47	1260	10.29	1260	0.00	12.64	1260	0.00	5.67
i25	1376	25.43	1376	34.62	1376	13.45	1376	0.00	12.62	1376	0.00	7.13
i26	1318	23.98	1318	35.62	1318	14.03	1318	0.00	12.62	1318	0.00	7.44
i27	1261	22.82	1261	33.73	1261	10.06	1261	0.00	12.64	1261	0.00	6.16
i28	1359	28.01	1359	42.90	1359	12.07	1359	0.00	12.71	1359	0.00	11.52
i29	1280	21.34	1280	31.98	1280	10.82	1280	0.00	12.62	1280	0.00	8.11
i30	1344	29.84	1344	36.71	1344	13.31	1344	0.00	12.58	1344	0.00	7.13
		27.16		36.28		12.99			12.76			8.17



Computational Results

Instances from Lalla et al. (2012); joint work with Eduardo Lalla-Ruiz

	GSPP		POPMUSIC-G r = 1, 2 ,3, 4		POPMUSIC r = 1, 2 ,3, 4		T2S+PR		
	opt.	t(s.)	obj. val.	t(s.)	obj. val. ,	t(s.)	best	gap (%)	t(s.)
40x5-01	2301	41.51	2301	53.07	2301	166.46	2303	0.09	0.90
40x5-02	2829	59.89	2829	55.23	2829	118.72	2834	0.18	1.09
40x5-03	2880	99.20	2880	59.06	2880	116.76	2880	0.00	0.50
40x7-03	—	—	2119	62.09	2119	122.78	2119	0.00	1.17
55x5-03	—	—	5499	106.71	5499	371.92	5499	0.00	2.67
55x5-05	—	—	5478	179.89	8478	299.65	5478	0.00	2.73
55x5-06	—	—	5595	79.09	5595	433.00	5595	0.00	2.56
55x7-03	—	—	3825	129.37	3825	196.18	3833	0.21	5.57
55x7-05	—	—	3797	151.00	3797	337.76	3801	0.11	3.56
55x7-06	—	—	3783	99.51	3783	194.23	3789	0.16	3.70
60x5-03	—	—	6780	123.21	6780	502.56	6780	0.00	4.25
60x5-06	—	—	6616	185.56	6616	418.88	6616	0.00	3.53

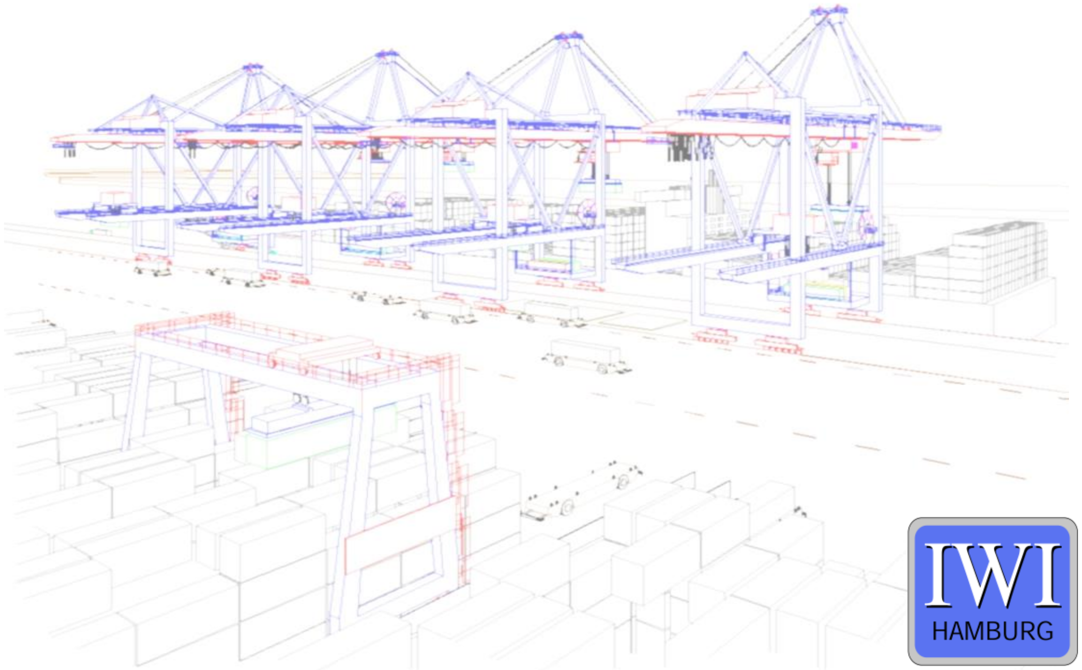
Conclusions and Future Research

- The DBAP has been solved by using *POPMUSIC*, it allows to work with a reduced problem and solve it to optimality.
- The use of an exact approach embedded in POPMUSIC constitutes a *matheuristic*.
- POPMUSIC provides new best objective function values for some benchmark instances (proposed by Lalla et al.)

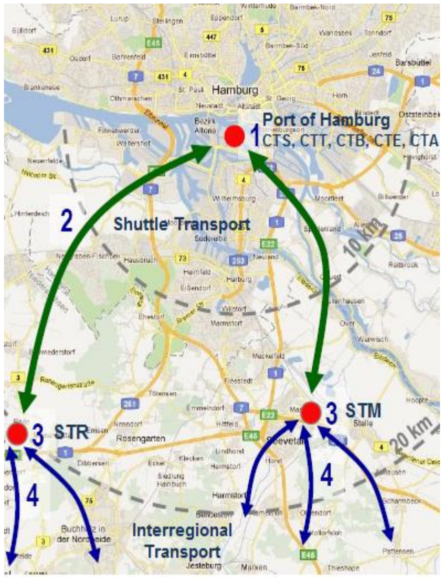
- Extending the DBAP to include different scenarios and time constraints.
- Analyze and implement an adaptive mechanism for the unique parameter of POPMUSIC, *r*, in order to reach a suitable balance between diversification and intensification along the search.
- In general: Extend the design choice towards the interplay of algorithm choice within POPMUSIC (exact vs. heuristic) with the size and number of parts. Integrate parameterization of the algorithm (like CPLEX for GPSS) itself.

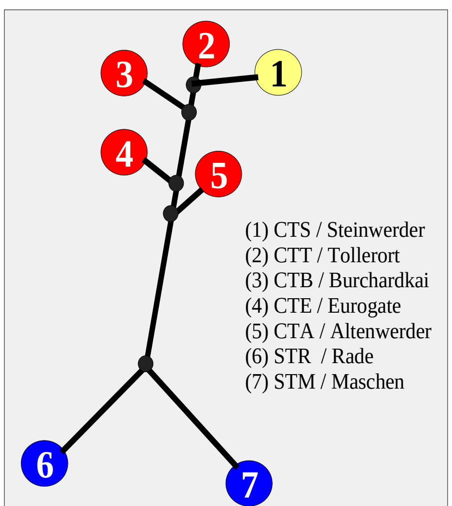
Extensions

A Research Talk



Hamburg


Hamburg
Port of Hamburg
1 CTS, CTT, CTB, CTE, CTA
2
Shuttle Transport
10 km
3 STR
3 STM
4
4
20 km
Interregional Transport


1
2
3
4
5
6
7
(1) CTS / Steinwerder
(2) CTT / Tollerort
(3) CTB / Burchardkai
(4) CTE / Eurogate
(5) CTA / Altenwerder
(6) STR / Rade
(7) STM / Maschen

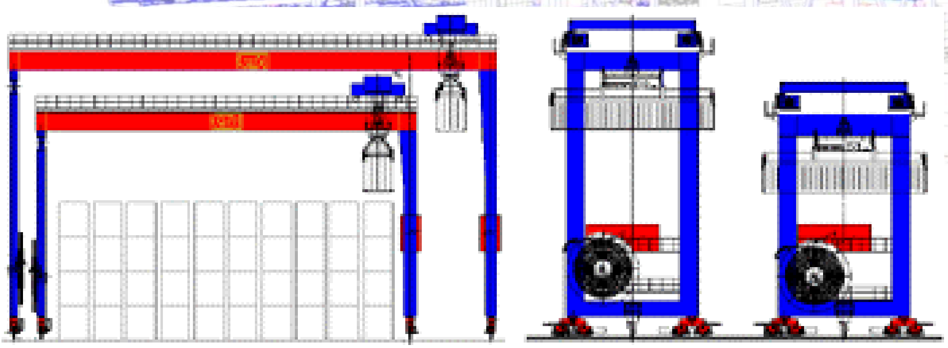

DUSKE & DUSKE
HAMBURG'S PORT
Finest Reserve
20% vol.
e75cl
DUSKE & DUSKE
HAMBURG'S PORT
Finest Reserve

Double Rail Mounted Gantry Cranes (DRMG)

- Allow for stacking 4-10 containers high
- Technical performance: ~ 20 moves/h

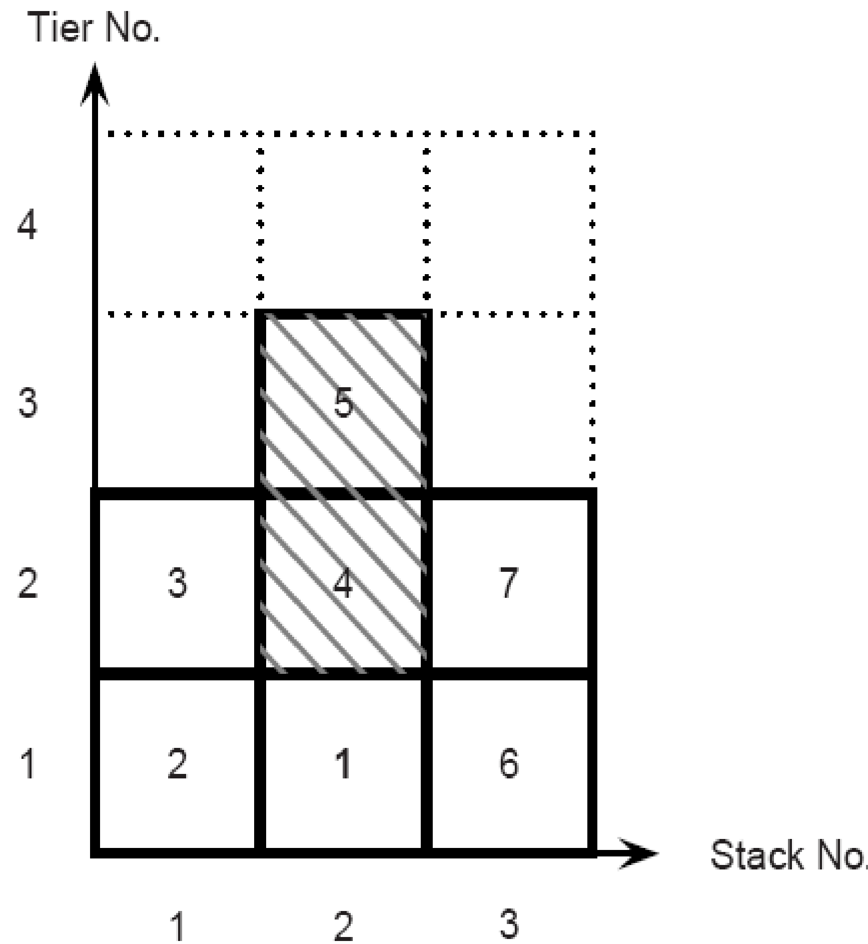
- Net productivity ??

- The more we utilize the space the more we need to account for making the right decisions where to locate containers. That is, for bad locational decisions we might have to relocate.



The Blocks Relocation Problem (BRP)

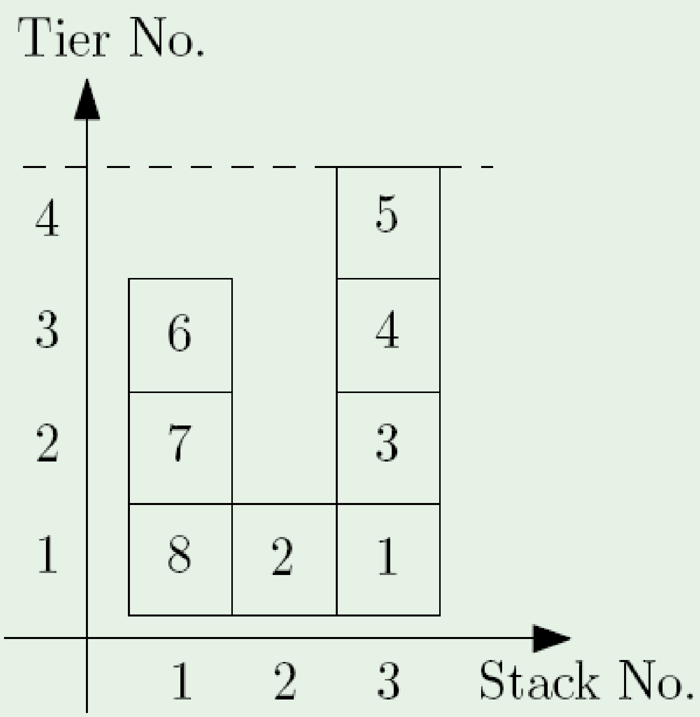
- <u>Given</u>
- A bay with *n* blocks which have to be retrieved
- <u>Assumptions</u> (Kim and Hong, 2006
- Retrieval order is given (1, 2, …, n)
- LIFO
- Only the upper blocks may be moved
- No pre-marshalling
- <u>Objective</u>
- Retrieval order such that the number of relocations is minimized



The Blocks Relocation Problem (BRP)

Assumptions discussion

- Assumptions (Kim and Hong, 2006)
- Only the upper blocks may be moved
- No pre-marshalling



Four relocations are needed to clear the stacking area:
Move block 2 first to stack 1.

But: This is forbidden by assumption.

With assumption: Six relocations are needed.

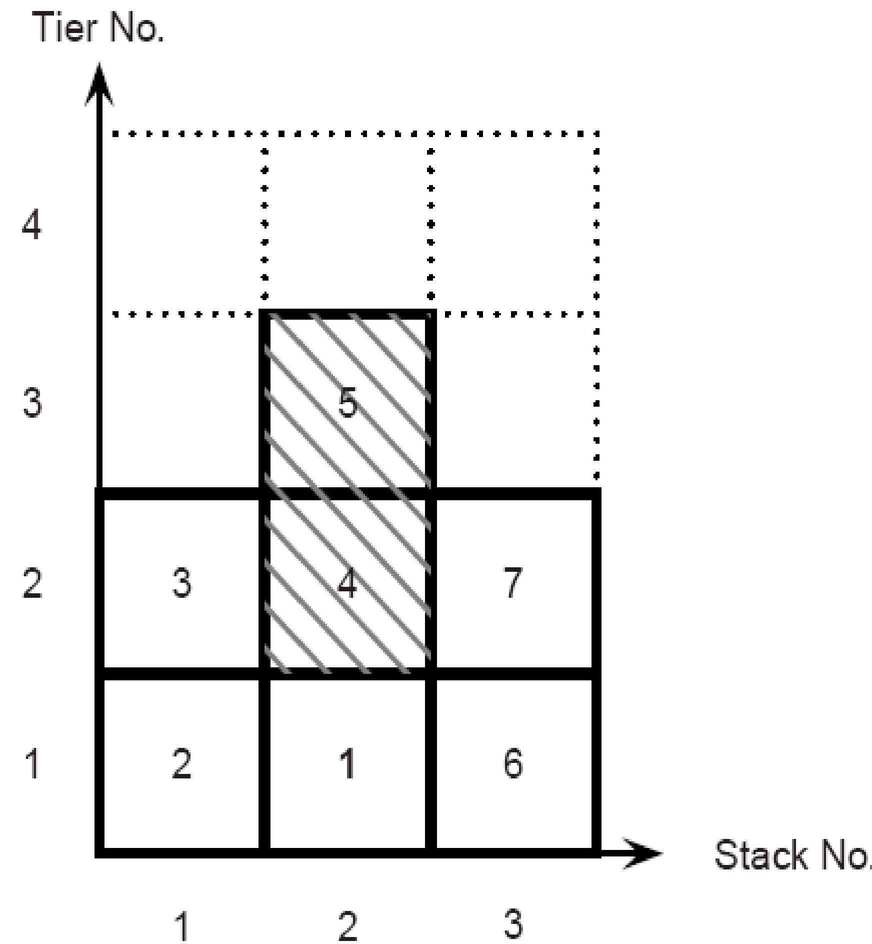
The Blocks Relocation Problem (BRP)

Dynamic Programming Formulation

- State variable: s = (k, i, t,C)

- k ∈ {1, . . . , n} is the block to be retrieved
- i ∈ {1, . . . ,m} is the stack in which the target block is found
- t is the list of blocks above the target block
- C is the configuration of the remaining blocks

- (Example: k = 1, i = 2, t = {5, 4}, an
- C = {{3, 2} , {7, 6}})

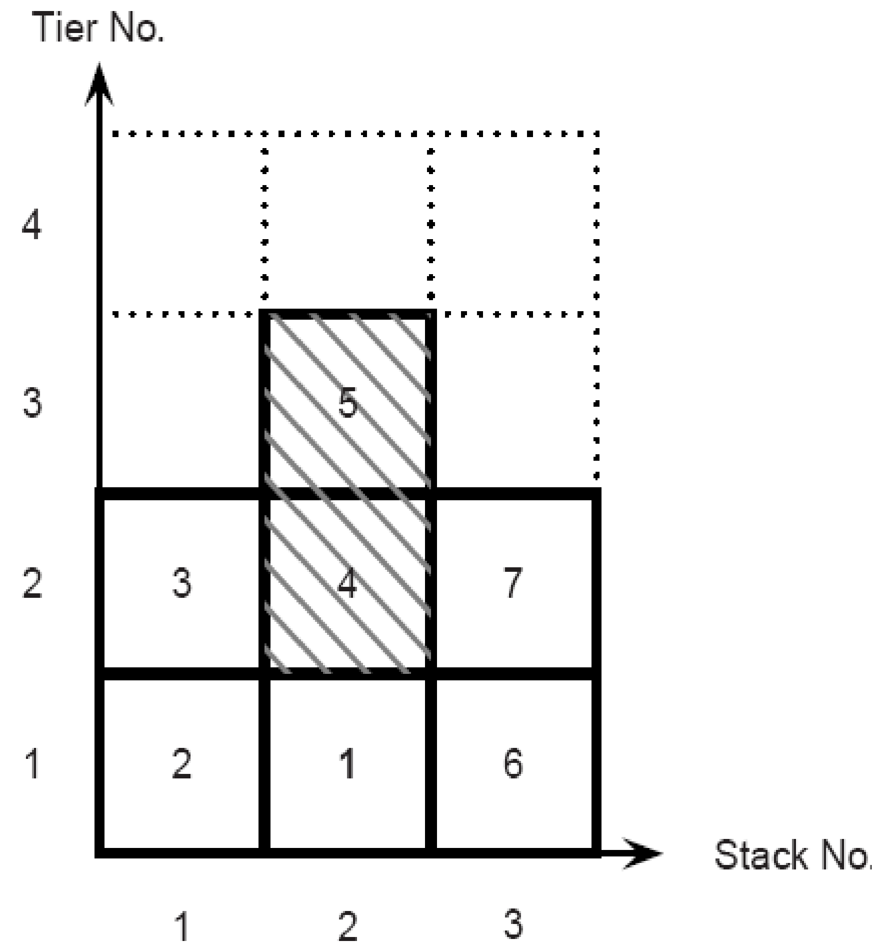


Dynamic Programming Formulation

- State transition function T(s, x):
- Let s′ = (k′, i′, t′,C′) be the state obtained by applying decision $x \in D(s)$ to the current state s.
- s′ = T(s, x)

(i) t = ∅: k′ = k + 1, i′ is the stack in which block k+1 is currently located, t′ is the list of blocks above k + 1 and C′ = C is the configuration of the remaining blocks

(ii) k′ = k, i′ = i, t′ = t \ {τ}, and C′ depends on the application of move x to block τ (e.g., with respect to the figure let us suppose that x = 1 → s′ = T(s, 1) = (1, 2, {4} ,C′), where C′ = {{5, 3, 2} , {7, 6}})



Dynamic Programming

State Variable : $s = (k, i, t, C)$, with k target block, i stack of target block, t list of blocks above the target block, and C configuration of remaining blocks;

Decision Variable : if τ is the uppermost block in sequence t, x indicates which stack block τ is moved to ($D(s)$ is the set of all feasible values of x w.r.t. the current state s);

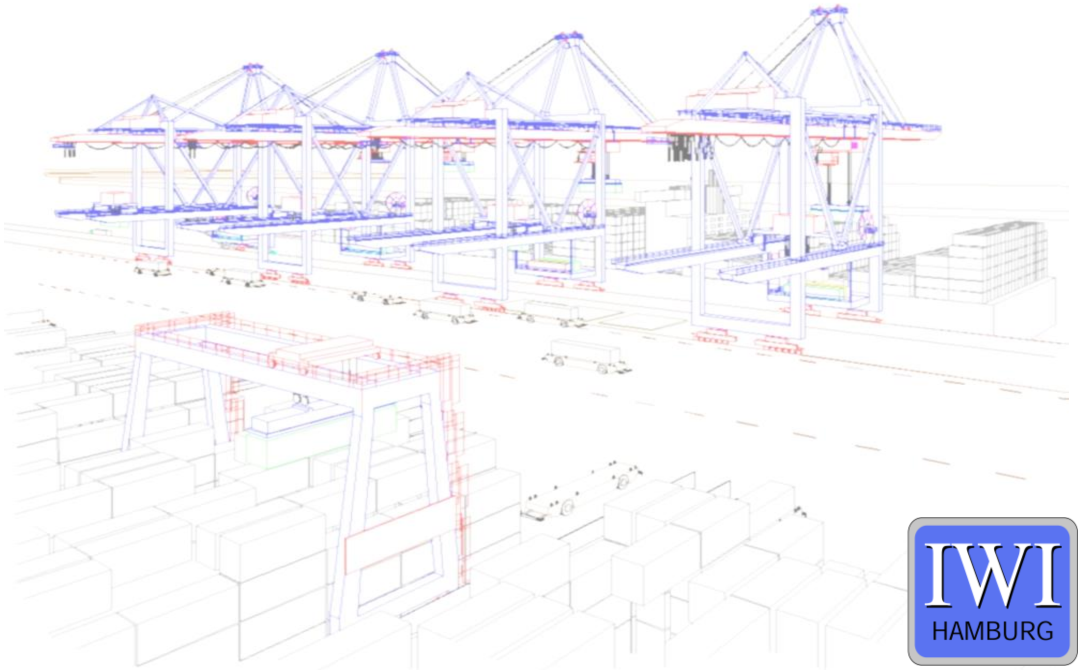
State Transition Function : a function T such that $s' = (k', i', t', C')$ is the state obtained by applying decision $x \in D(s)$ to the current state s, which is, $s' = T(s, x)$;

Functional Equation : DP "backward" functional equation

$$f(k,i,t,C) = \begin{cases} 1 + f\big(k+1, i', t', C\big), & t = \emptyset, \\ 1 + \min\limits_{x \in D(k,i,t,C)} \Big\{ f\big(k, i, t \setminus \{\tau\}, C'\big) \Big\}, & t \neq \emptyset, \end{cases}$$

with $f(n, i, \emptyset, C) = 1$.

Interlude



IWI HAMBURG

Move-based vs. Method-based Neighborhoods

(Paradigms?)

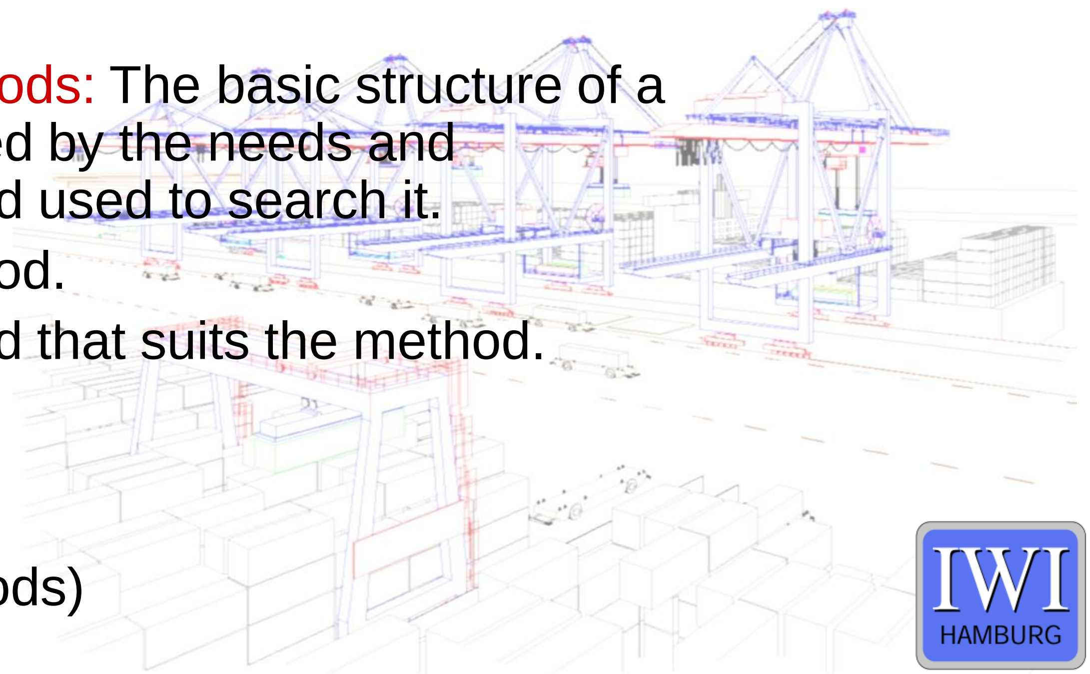
Move-based neighborhoods: Neighborhood search based on topological concepts relating a given solution to similar solutions via local changes or moves. (Example: small homogeneous neighborhoods, very large scale neighborhoods)

1. Define a neighborhood
2. Find efficient methods for its exploration

Method-based neighborhoods: The basic structure of a neighborhood is determined by the needs and requirements of the method used to search it.

1. Assume a given method.
2. Define a neighborhood that suits the method.

(Model-based neighborhoods)



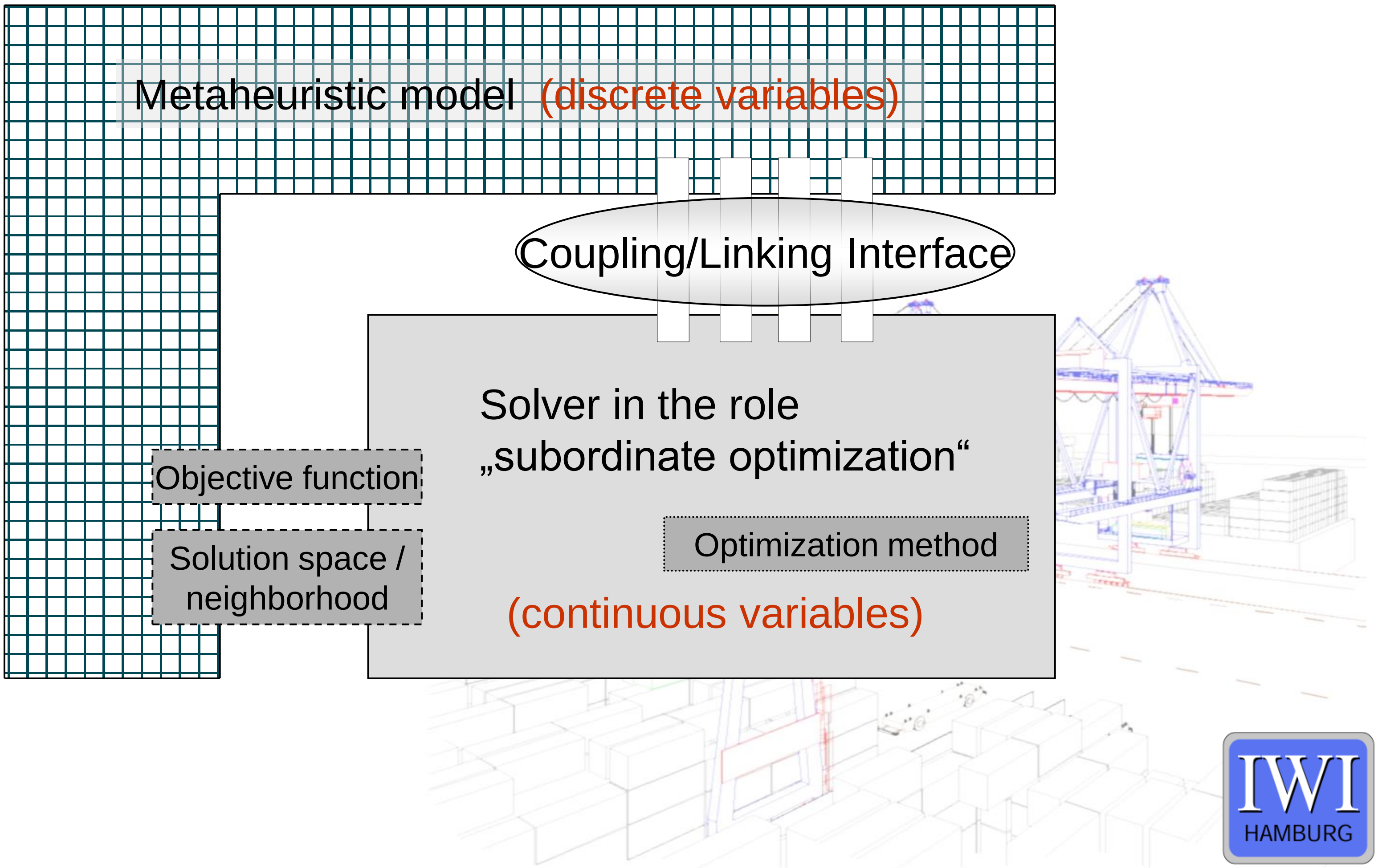
The Corridor Method (CM) Sniedovich/Voß (2006)

The CM can be described as a local search method where neighborhoods are relatively large sets whose structure and size are compatible with the optimization method operating on it.

- The optimization problem under consideration is *large*.
- There is an optimization method for efficiently solving smaller instances of the problem.
- It is easy to generate (an initial) feasible solution.
- There is an efficient method for generating suitably large neighborhoods around feasible solutions to the problem on which the optimization method can be used.

M. Sniedovich, S. Voß. The Corridor Method: a Dynamic Programming Inspired Metaheuristic. *Control and Cybernetics*, 35(3):551-578, 2006.

Metaheuristic and Optimization

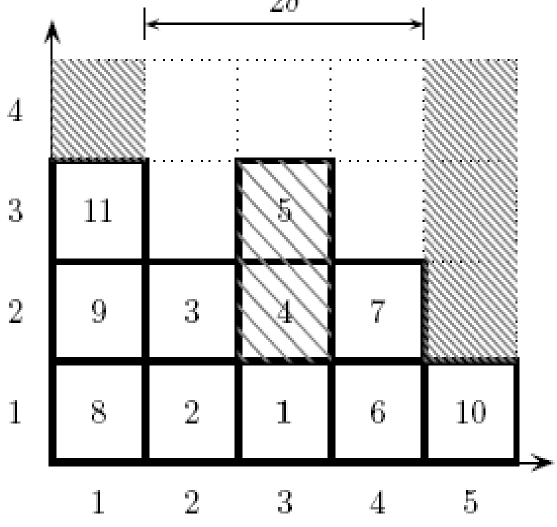


•**Matheuristics** are made by the interoperation of metaheuristics and mathematical programming (MP) techniques. An essential feature is the exploitation in some part of the algorithms of features derived from the mathematical model of the problems of interest, thus the definition "*model-based metaheuristics*" appearing in the title of some events of the conference series dedicated to matheuristics.

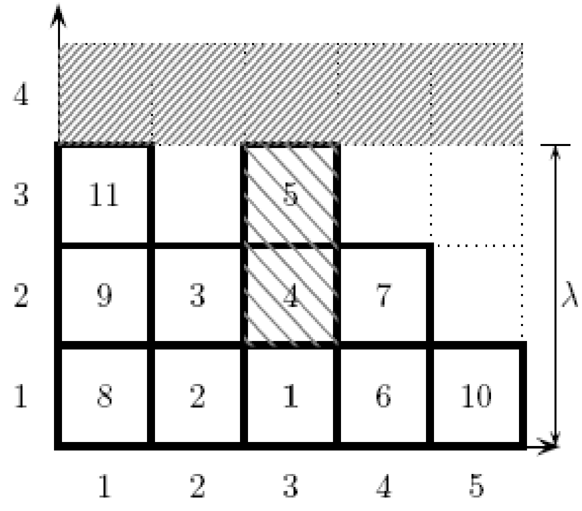
•Matheuristics. Annals of Information Systems, Vol. 10. Maniezzo, V., Stützle, T., Voß, S. (Eds.), Springer, 2009.

•Special Issue on Matheuristics. Guest Editors: K. Dörner, V. Maniezzo, S. Voß, and P. Hansen, Discrete Applied Mathematics, to appear (2013)

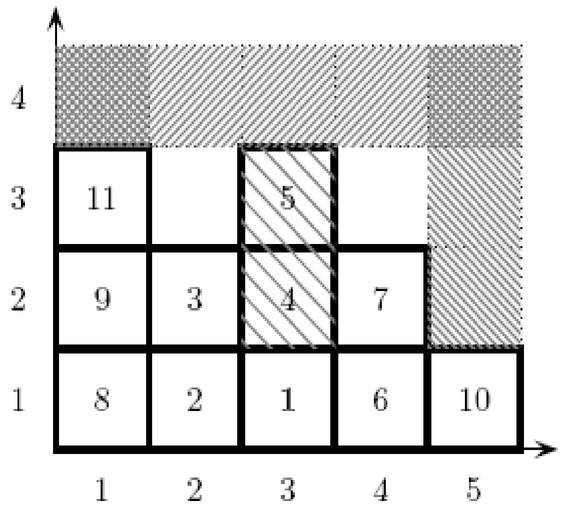
The Corridor



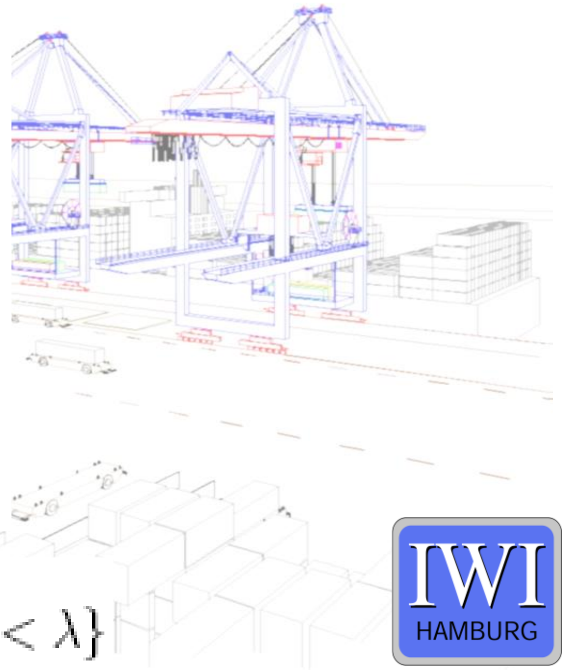
(a)



(b)



(c)



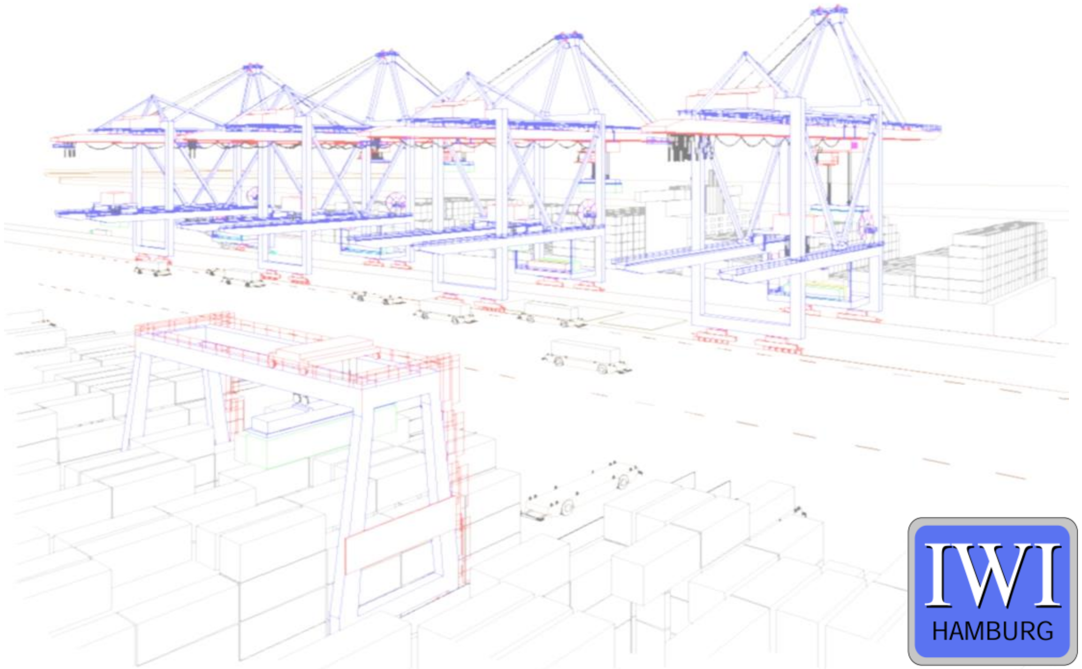
$$D(s, \delta, \lambda) = \{x \in \{1, \ldots, m\} \setminus \{i\} : i - \delta \leq x \leq i + \delta,\ |c_x| < \lambda\}$$

Numerical Results

Bay Size		KH		CM		Corridor	
h	m	No.	Time†	No.	Time†	δ	λ
5	4	23.7	0.1	16.6	0.5	2	6
5	5	37.5	0.1	18.8	0.8	2	6
5	6	45.5	0.1	22.1	0.8	2	6
5	7	52.3	0.1	25.8	1.43	1	7
5	8	61.8	0.1	30.1	1.46	1	6
5	9	72.4	0.1	33.1	1.41	1	6
5	10	80.9	0.1	36.4	1.87	1	6

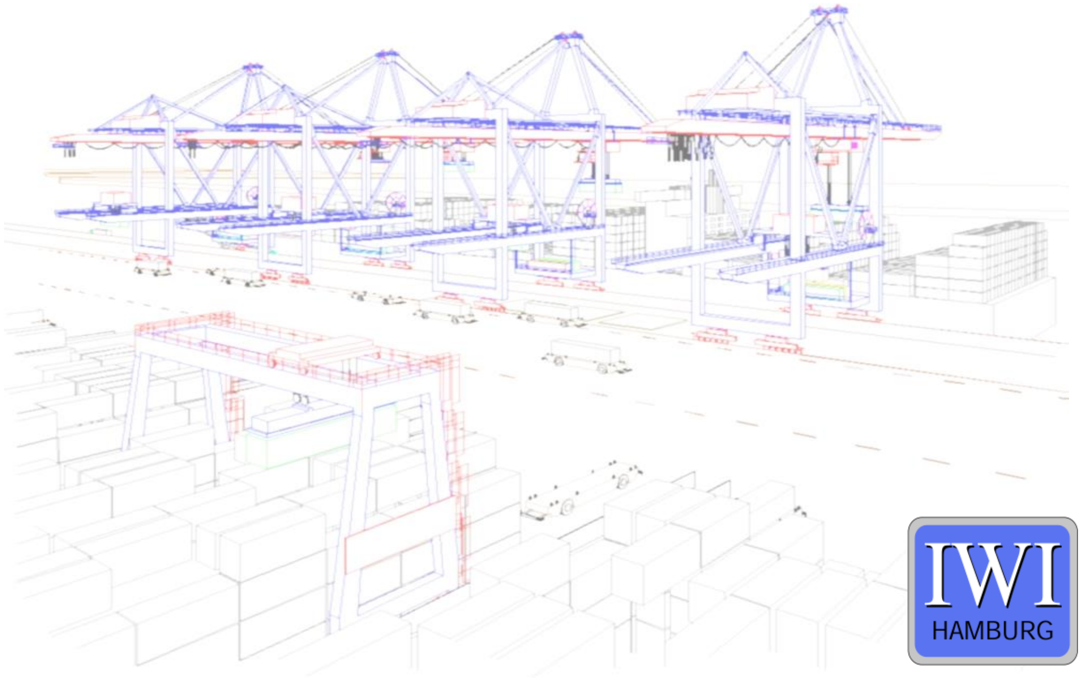
† : CPU seconds on a Pentium-IV 512Mb RAM.

A New Talk ?



Promising solution to avoid congestion in seaports

BTW: Vehicle Routing Problems have a natural representation in terms of the POPMUSIC frame.



Questions? Answers? Discussion

BTW: Vehicle Routing Problems have a natural representation in terms of the POPMUSIC frame.

Post them towards

stefan.voss@hamburg.de ☺